

Graduado en Ingeniería Informática

Universidad Politécnica de Madrid

Facultad de Informática

TRABAJO FIN DE GRADO

Desarrollo de una aplicación Web empresarial
de ayuda a la producción de campañas
publicitarias

Autora: Marjorie Aydeé Ganchala Mora

Tutor: Rafael Fernández Gallego

MADRID, ENERO DE 2015

A mi abuela María y a mis queridos padres.

AGRADECIMIENTOS

Durante el desarrollo de este trabajo, de fin de carrera son varias las personas a las que debo mi más sincero y profundo agradecimiento.

A mi familia, en especial a mi abuela y a mis queridos padres que han sido el mejor ejemplo de superación, gracias a su esfuerzo y sacrificio he podido alcanzar este primer objetivo. Siempre han estado a mi lado apoyándome, dándome ánimo y aconsejándome en los momentos más difíciles de mi carrera.

A mi tutor académico Rafael Fernández Gallego que me ha apoyado para realizar este trabajo y me ha tenido una inmensa paciencia.

A mis tutores profesionales Daniel Paz y Jorge Lázaro responsables del departamento de sistemas de Canalmail S.L, que confiaron en mi para llevar a buen término este proyecto y a todos los implicados de alguna manera en la realización de este proyecto, gracias por vuestra inestimable ayuda.

También me gustaría agradecer a mis queridos amigos, con los que he compartido las mejores experiencias en estos últimos años de carrera, gracias por su colaboración, su amistad y su apoyo incondicional.

¡A todos mil gracias!

Índice

1.	ENTORNO DE TRABAJO.....	1
1.1.	La empresa.....	2
2.	RESUMEN	1
2.1.	Español.....	2
2.2.	Inglés.....	4
3.	INTRODUCCIÓN.....	6
3.1.	Antecedentes del proyecto.....	7
3.2.	Descripción del proyecto.....	10
3.3.	Objetivos.....	11
3.4.	Fases del desarrollo.....	11
3.5.	Dificultades encontradas.....	13
3.6.	Estructura de la memoria final.....	14
4.	ESTADO DEL ARTE	16
4.1.	Frameworks.....	17
4.1.1.	Symfony2 [10].....	17
4.1.2.	Zend Framework. [12].....	18
4.1.3.	Comparación.....	19
4.1.4.	Decisión final.....	20
4.2.	Tecnologías actuales	21
4.2.1.	HTML5 [17].....	21
4.2.2.	JavaScript [18].....	22
4.2.3.	Framework CSS.....	25
4.2.4.	PHP [31].....	27
4.2.5.	Python [35].....	29
4.3.	Algoritmos.....	30
4.3.1.	Algoritmo de búsqueda: Boyer-Moore. [39]	30
4.3.2.	Algoritmo Knuth-Morris-Pratt [41].....	31
4.3.3.	Decisión del algoritmo.	31
4.4.	Base de datos.....	32
5.	DESARROLLO.....	33

5.1.	Requisitos.....	34
5.1.1.	Lista de requisitos.....	34
5.2.	El sistema.....	36
5.2.1.	Componentes del proyecto.	36
5.3.	Sistema de usuarios.....	38
5.3.1.	Usuarios finales	38
5.3.2.	Datos necesarios para acceder al sistema.	39
5.4.	Función principal: Programación de un formulario.....	41
5.4.1.	Carga y visualización del formulario.	41
5.4.2.	Módulo analizador.....	41
5.4.3.	Selección de las validaciones y BBDD.	41
5.4.4.	Módulo modificador.....	41
5.4.5.	Comprobación del formulario.	41
5.5.	Carga y visualización del formulario.....	42
5.5.1.	Visualización	43
5.6.	Módulo analizador.....	46
5.6.1.	Funcionamiento del script.	46
5.7.	Selección de las validaciones.....	50
5.7.1.	Clasificación.....	50
5.7.2.	Lista de validaciones.	51
5.7.3.	Enganchar las validaciones.....	55
5.8.	Módulo modificador.....	57
5.8.1.	Funcionamiento del script:	57
5.9.	Comprobación del formulario.....	61
5.9.1.	Prueba del funcionamiento.....	61
5.10.	Sistema de Gestión de bases de datos.	62
5.10.1.	Envío de los datos a la BBDD.	62
5.11.	Gestión de formulario.....	63
6.	VISUALIZACION	64
6.1.	Vistas.	65
6.1.1.	Iniciar sesión y registro.....	65
6.1.2.	Menú principal.....	66
6.1.3.	Subir Formulario	66

6.1.4.	Visualización del formulario.	67
6.1.5.	Selección de las validaciones.	67
6.1.6.	Fin de la validación	69
6.1.7.	Cambiar contraseña	70
6.1.8.	Lista de los formularios.	70
7.	PRUEBAS	71
7.1.	Pruebas.....	72
7.1.1.	Prueba 1: Crear una cuenta.	72
7.1.2.	Prueba 2: Descargarse el manual.....	72
7.1.3.	Prueba 3: Diseño de una plantilla.....	72
7.1.4.	Prueba 4: Programar el formulario.	72
8.	CONCLUSIONES.....	73
8.1.	Conclusión.....	74
9.	FUTURAS LINEAS DE DESARROLLO	75
9.1.	Mejoras.	76
9.1.1.	Volver a modificar un formulario.....	76
9.1.2.	Mejorar el Front-End de la aplicación.....	76
9.1.3.	Añadir las validaciones restantes.....	76
9.1.4.	Crear un panel de administrador.....	76
10.	BIBLIOGRAFÍA	77
10.1.	Bibliografía.....	78
A.	ANEXO	82
A.1.	Manual de usuario.....	83
A.1.1.	Registro e inicio de sesión.....	83
A.1.2.	Subir Formulario.	84
A.1.3.	Visualización del formulario.	84
A.1.4.	Selección del País para las validaciones.....	85
A.1.5.	Validaciones y campos de la BBDD.	85
A.1.6.	Configuración.....	86
A.1.7.	Ver formularios.	86
A.2.	Campos de la BBDD.....	87
A.2.1	Esquema de la BBDD.....	87
A.3.	Tipo de validaciones.	89

Índice de figuras

Figura 1: Logotipo de la empresa Media Response Group.	2
Figura 2: Logotipo de HOTWords.	2
Figura 3: Logotipo de Webzodes.....	2
Figura 4: Logotipo de Efficient Target.....	3
Figura 5: Logotipo de Canalmail S.L.	3
Figura 6: Esquema básico del funcionamiento de una campaña.	2
Figure 7: Basic scheme of the operation of a campaign.....	4
Figura 8: Flujo de trabajo básico para la creación de una campaña.	7
Figura 9: Java Struts. [8]	8
Figura 10: Repositorio de todas las campañas creadas.	8
Figura 11: Ejemplo de los ficheros de una campaña.	9
Figura 12: Ejemplo de una campaña publicitaria.	10
Figura 13: Arquitectura Modelo Vista Controlador. [9].....	17
Figura 14: Logotipo del Framework Symfony. [11]	17
Figura 15: Logotipo del Framework Zend. [13].....	18
Figura 16: Volumen de búsqueda Symfony vs Zend. [14].....	19
Figura 17: Logotipo del IDE Netbeans. [16].....	19
Figura 18: Logotipo de HTML5 [17].	21
Figura 19: Logo de JavaScript. [19]	22
Figura 20: Logotipo de jQuery. [20]	22
Figura 21: Logotipo de Mootools. [21]	22
Figura 22: Comparación de búsqueda entre jQuery y Mootools. [24]	23
Figura 23: Diferencia entre jQuery y Mootools. [25].....	23
Figura 24: Logotipo de Bootstrap. [26]	25
Figura 26: Logotipo de PHP. [32]	27
Figura 27: Icono de Java Server Pages. [33]	27
Figura 28: Diferencias entre PHP y JSP. [34]	28
Figura 29: Logotipo del lenguaje de programación Python. [36]	29
Figura 30: Logotipo de Perl. [37]	29

Figura 31: Ejemplo de aplicación del algoritmo Boyer-Moore. [40]	30
Figura 32: Aplicación del algoritmo KMP.	31
Figura 34: Departamento creativo de la empresa. [47]	38
Figura 35: Pantalla de registro.....	39
Figura 36: Pantalla para iniciar sesión.....	40
Figura 37: Menú al acceder al panel.....	42
Figura 38: Cargando un nuevo formulario.	43
Figura 39: Ejemplo de visualización de un formulario ficticio.	44
Figura 40: Vista de la aplicación mostrando los errores habituales.	45
Figura 41: Mensaje con la solución de un error.	45
Figura 42: Flujo de trabajo del script Analizador.....	46
Figura 43: Segmentación del diseño del formulario.....	47
Figura 44: Etiqueta de comentario en HTML5.	47
Figura 45: Etiqueta de inicio del código HTML5.	47
Figura 46: Etiquetas de los principales bloques del formulario.	48
Figura 47: Etiquetas dentro del bloque cuerpo del documento HTML5.	48
Figura 48: Estructura de un formulario con los campos que se van a procesar.	49
Figura 49: Contenido del fichero Input_Parser.html.	49
Figura 50: Ejemplo de selección de tipo de validación.	55
Figura 51: Ejemplo de selección de campo de base de datos.....	56
Figura 52: Flujo de trabajo del script Modificador.....	57
Figura 53: Ejemplo de validación de los campos de un formulario.	58
Figura 54: Parámetros que recibe el script Modificar.	58
Figura 55: Ejemplo de la información que contienen los parámetros	59
Figura 56: Ejemplo de los campos del formulario ya modificado.	59
Figura 57: Funcionamiento de la validación de los campos.....	60
Figura 58: El ejemplo en formato JSON.	62
Figura 59: Opciones del menú principal.	63
Figura 60: Lista de los formularios guardados en el directorio del usuario.	63
Figura 61: Vista de la página principal, siendo iniciar sesión.	65
Figura 62: Vista del formulario para el registro.	65
Figura 63: Vista del menú principal.	66
Figura 64: Vista de subir formulario.	66

Figura 65: Vista de un formulario ficticio cargado correctamente.....	67
Figura 66: Vista de la selección del país de la campaña.	67
Figura 67: Vista de un formulario preparado para programarse.	68
Figura 68: Lista de los tipos de validación del formulario.	68
Figura 69: Lista de los campos de la base de datos.	69
Figura 70: Vista de las opciones finales de un formulario.	69
Figura 71: Vista de cambiar la contraseña.....	70
Figura 72: Vista de la lista de formularios.	70
Figura 73: Página de inicio.....	83
Figura 74: Menú del usuario.....	83
Figura 75: Servicio para cargar el nuevo formulario.....	84
Figura 76: Visualización del formulario.....	84
Figura 77: Selección del país para las validaciones.	85
Figura 78: Seleccionar las validaciones y los campos de la BBDD.....	85
Figura 79: Finalización de las validaciones.....	86
Figura 80: Configuración.	86
Figura 81: Lista de formularios programados.	86

Índice de tablas

Tabla 1: Ejemplo de la aplicación del algoritmo KMP	31
Tabla 2: Patrón telefónico según el país de teléfonos fijos [43].....	52
Tabla 3: Patrón telefónico según el país de teléfonos móviles [44]	52
Tabla 4: Tipo de codificación ISO dependiendo del país. [45]	53
Tabla 5: Patrón del código postal dependiendo del país.	54
Tabla 6: Ejemplo de los datos que se envían.....	62

1. ENTORNO DE TRABAJO

1.1. La empresa.

Este proyecto se realiza para Canalmail S.L. que forma parte del corporativo Media Response Group.



Figura 1: Logotipo de la empresa Media Response Group.

Media Response Group [1] es un grupo de compañías que se especializan en el mercado de comunicación online, fundada en 1999. Se especializa en ofrecer al anunciante el cumplimiento de sus objetivos a través del producto y la tecnología más adecuada a sus necesidades. Dicho grupo se compone de las siguientes compañías:



Figura 2: Logotipo de HOTWords.

HOTWords [2] fue fundada en 2006 y se integró a Media Response Group en 2009. Es una empresa dedicada a la *publicidad contextual*¹. Su fuerte reside en la *capacidad de segmentación*², el *engage*³ del usuario con los formatos y en la notoriedad de los resultados.



Figura 3: Logotipo de Webzodes.

¹ **Publicidad contextual**, Publicidad que aparece en sitios web o blogs que tienen relación con el tema o palabras claves que se tratan en dichos sitios.

² **Capacidad de segmentación**, consiste en conocer a los consumidores para adecuar el producto o servicio de las características del mercado.

³ **Engage**, este término se refiere a la interrelación entre la marca publicitaria y el público al que esta destinado.

Webzodes [3] es una plataforma alternativa para la difusión de video publicitario online a través de un formato de alto impacto y gran cobertura que nace a finales de 2012. La finalidad de esta plataforma es asegurar la calidad de la marca publicitaria manteniendo el respeto hacia los usuarios, para esto se desarrollan piezas creativas con un gran impacto visual para conseguir altos niveles de *CTR*⁵, en inglés “*Click Through Rate*”.



Figura 4: Logotipo de Efficient Target.

Efficient target [4] es una plataforma orientada al marketing digital de resultados permitiendo al anunciante mitigar su riesgo y optimizar sus costes. Esta tecnología optimiza las campañas en tiempo real ya que dispone de un algoritmo *behavioral targeting*⁶ que es capaz de generar perfiles de usuario después de analizar las cookies y de este modo obtener una mejor respuesta ante el mensaje que se envía, de esta forma combinando el *retargeting*⁷ con la calidad de la marca publicitaria, se llega al objetivo final de cada acción y campaña para la mayoría de los anunciantes.



Figura 5: Logotipo de Canalmail S.L.

Canalmail S.L [5] es una empresa dedicada al envío de email marketing, siendo en la actualidad el líder y claro referente en esta actividad dentro del país. Gestiona la calidad de la marca publicitaria y performance marketing⁸ dentro de la campaña según las necesidades del anunciante, gracias a su amplia lista de segmentación, siempre cuidando los intereses de los usuarios finales.

Dada la gran demanda de envío de campañas publicitarias, surge la necesidad de realizar este proyecto cuyo objetivo principal es agilizar el desarrollo de una campaña, en otras palabras, optimizar el tiempo que se dedica a la programación de los formularios online.

⁵ **CTR**, es una métrica que mide el interés y la eficiencia de una campaña publicitaria.

⁶ **Behavioral-targeting**, algoritmo de segmentación de comportamiento de usuarios.

⁷ **Retargeting**, es una técnica de publicidad online que recupera la información de las cookies de posibles consumidores finales.

⁸ **Performance marketing**, es un modelo de pago de marketing online, en la que el anunciante paga sólo por los objetivos alcanzados.

Después de realizar mi prácticum en Canalmail, y conocer todos los detalles del funcionamiento de esta empresa, en concreto todo lo relacionado con la creación y lanzamiento de las campañas publicitarias, los tutores profesionales me han asignado la tarea de crear una aplicación para agilizar el proceso de lanzamiento.

2. RESUMEN

2.1. Español.

La creación de esta aplicación web empresarial surge con la necesidad de optimizar el tiempo en el proceso de creación de una campaña publicitaria de email marketing.

El objetivo principal de este trabajo es automatizar el proceso de validación de los campos de un formulario web. Un formulario web [6] es un documento digital en el que los usuarios introducen sus datos personales como nombre, apellido, dirección, documento de identidad, entre otros.

Estos datos posteriormente serán procesados y almacenados en una base de datos para luego ser enviados al anunciante.

El proceso de validación se refiere a la programación del formulario web en la parte del cliente usando tecnologías web como JavaScript y HTML5, para controlar que los datos introducidos por el usuario en el formulario, sean correctos.

Cada campo de un formulario web tiene una validación específica que depende de varios factores, como son el país de lanzamiento de la campaña y el campo a validar. De esta forma dependiendo del tipo de validación se genera un fichero JavaScript con todas las validaciones de dicho formulario.

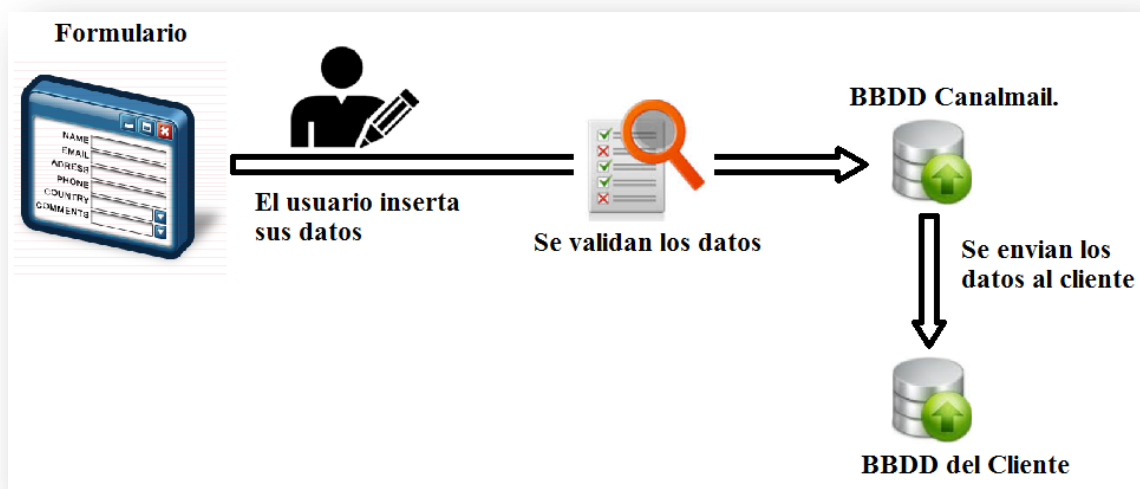


Figura 6: Esquema básico del funcionamiento de una campaña.

Una de las finalidades de este trabajo es que cualquier usuario de la empresa pueda programar un formulario web, sin tener conocimientos previos de programación, ya que la programación se realiza de forma transparente al usuario.

Este es un resumen básico de la aplicación web, sin embargo se debe tener en cuenta una serie de requisitos y parámetros para hacerlo más eficiente y personalizable dependiendo de las necesidades del producto final de cada campaña publicitaria. Todos estos aspectos se explicaran en detalle en los siguientes apartados.

Este trabajo se realizó en el corporativo Media Response Group, para la empresa Canalmail S.L, situada en Alcobendas, supervisado por los tutores profesionales Daniel Paz y Jorge Lázaro Molina y por el tutor académico Rafael Fernández Gallego de la Universidad Politécnica de Madrid.

2.2. Inglés.

The creation of this enterprise Web application arises from the need to optimize the time in the process of creating an online advertising campaign.

The main objective of this work is to automate the process of validating fields in a web form. A web form [6] is a digital document that users enter data such as name, surname, address, ID number among others. These data will subsequently be processed and stored in a database and then be sent to the client.

These data will subsequently be processed and stored in a database and then be sent to the advertiser.

This validation process refers to programming the online form on the client's side using web technologies such as JavaScript, HTML5 to control that the data entered by the user in this form are correct.

Each field in a web form has a specific validation that depends on several factors; like being a nationwide launch of the campaign and validating data, thus depending on the type of validation a JavaScript file is generated with all validation web form. This file is integrated into the web form by calling the service.

One purpose of this work is that any business user can program a web form, without prior knowledge of web programming, since programming is transparent to the user.

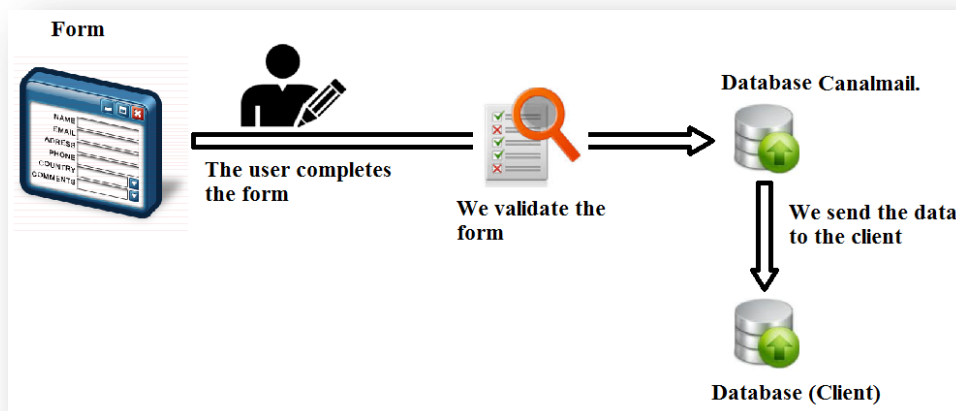


Figure 7: Basic scheme of the operation of a campaign.

This is a basic summary of the web application; however we must consider a number of requirements and parameters to make it more efficient and customizable depending on the needs of the end product of each advertising campaign. All these aspects are explained in detail in the following sections.

This work was performed in the corporate Media Response Group, for the company Canalmail S.L, located in Alcobendas, supervised by professional tutors Daniel Paz and Jorge Lázaro Molina and PhD Assistant Lecturer at *Universidad Politécnica de Madrid*. Rafael Fernández Gallego.

3. INTRODUCCIÓN

3.1. Antecedentes del proyecto.

Para entender mejor como funciona una campaña de email marketing, se explicará el proceso de creación básico:

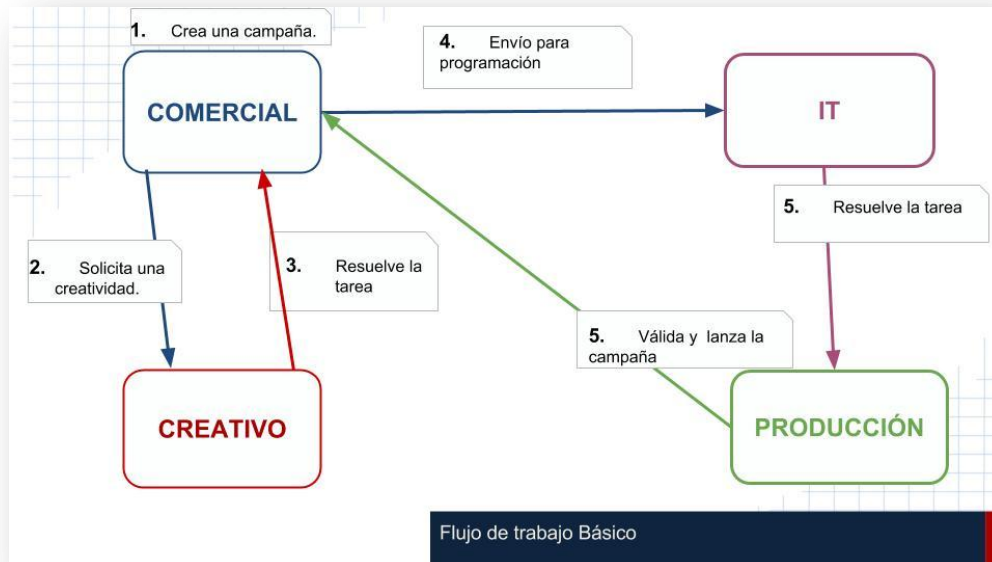


Figura 8: Flujo de trabajo básico para la creación de una campaña.

El comercial es la persona encargada de contactar con los clientes y negociar el lanzamiento de una determinada campaña publicitaria.

Cuando se crea una campaña el comercial pide al cliente una serie de requisitos previos necesarios para crear el diseño del formulario web. Esta información se envía al departamento creativo que se encarga de resolver la tarea.

Una vez aceptada la creatividad por parte del cliente, se envía el formulario web al departamento de IT para realizar la programación y verificar que los datos se insertan correctamente en la base de datos.

Cuando se termina el proceso de programación, se envía el formulario web programado al departamento de producción que se encarga de preparar el lanzamiento de la campaña.

Este diagrama resume de forma básica como interactúan los departamentos para el lanzamiento de la campaña, este escenario es en el mejor de los casos.

Sin embargo, si el cliente decide a última hora antes de lanzar la campaña añadir un campo nuevo al formulario o modificar alguno ya existente, se tiene que rehacer el formulario lo que conlleva volver a realizar la programación. Esta es la principal razón por la que se realiza este proyecto.

El sistema de validación de formularios que existía antes de la realización de este proyecto usaba Apache Struts [7], una framework basado en la arquitectura software MVC⁹ que da soporte a aplicaciones web en la plataforma *Java Enterprise Edition* (Java EE).



Figura 9: Java Struts. [8]

En la siguiente imagen se ve como es necesario crear manualmente cada carpeta con el código del pedido, siendo el identificador único que tiene cada campaña.

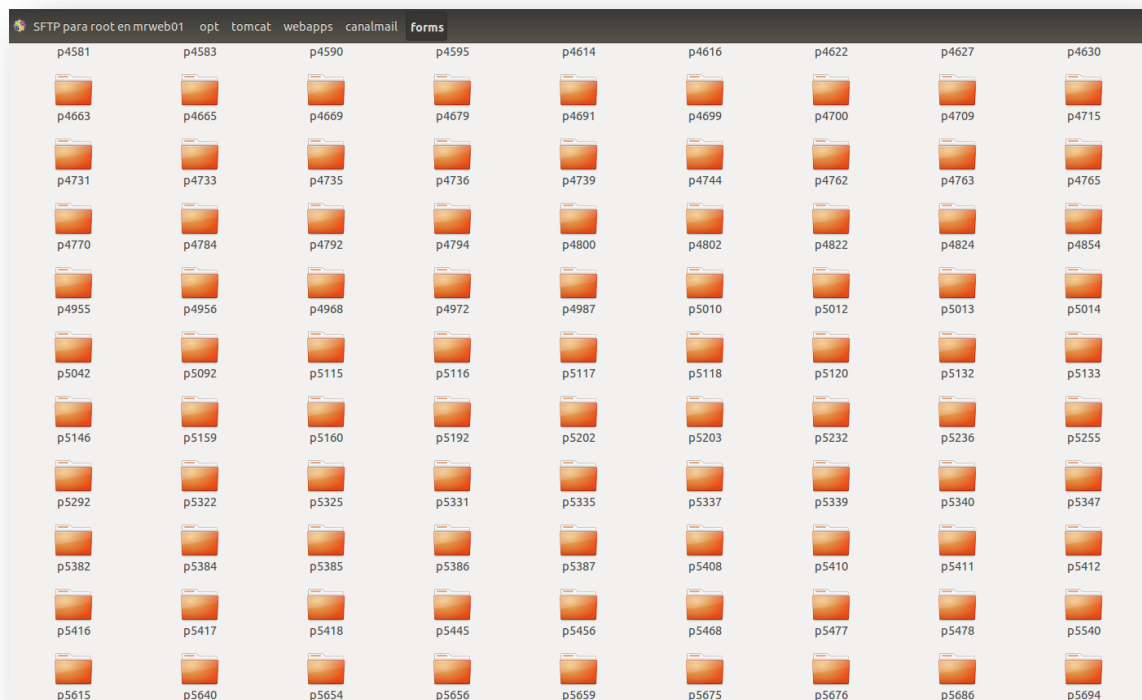


Figura 10: Repositorio de todas las campañas creadas.

⁹ MVC, en inglés *Model-View-Controller*. En castellano *Modelo Vista Controlador*, es un patrón de arquitectura software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Uno de los problemas de este sistema se debe a la sincronización que existe en el sistema de preproducción con el de producción por lo que si te equivocas al crear la carpeta no se puede borrar.

Dentro de cada carpeta se encuentran dos ficheros que son: “formulario.jsp” y “gracias.jsp”.

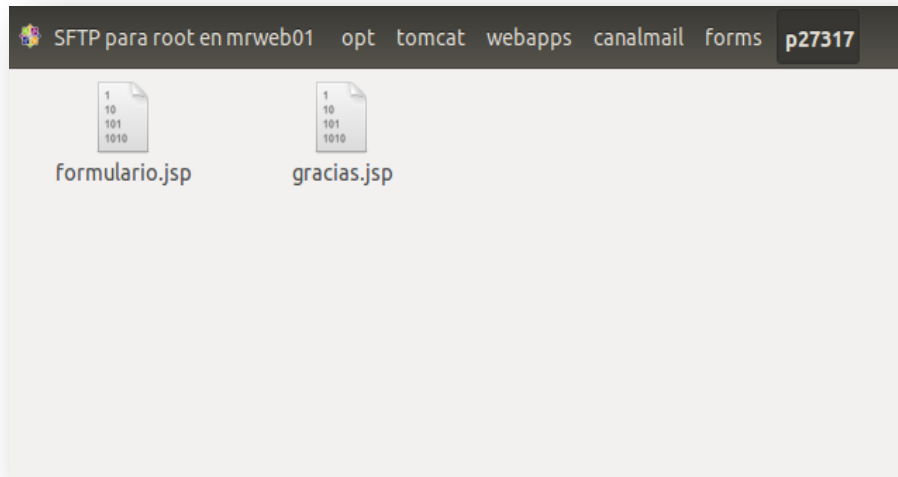


Figura 11: Ejemplo de los ficheros de una campaña.

Estos ficheros se crean manualmente copiando el diseño del formulario que ha sido creado anteriormente, añadiendo la programación manualmente uno a uno.

El mayor problema detectado en este sistema se da en cómo se ha programado la redirección de los formularios. Si un usuario hace clic en el botón de enviar del formulario, se genera una petición desde dicho formulario a la página de gracias.

Esta redirección muestra la página de gracias del anunciante con una URL incorrecta, esto afecta a las medidas de rendimiento que contabilizan el número de registros dentro de la empresa.

Otro inconveniente que se detecta en este sistema es que no existe documentación de la herramienta por lo que al intentar mejorar o cambiar algo de su estructura resulta tedioso dado que hay que leer miles de líneas de código dentro de varios paquetes enlazados entre sí.

Tras realizar el estudio comparativo entre crear un proyecto nuevo o intentar mejorar el existente, se decidió crear un nuevo proyecto dado que el tiempo a emplear es mucho menor y además se corregirán los errores observados.

3.2. Descripción del proyecto.

El proyecto se centra en la creación de una aplicación Web empresarial que ayude a la producción de campañas publicitarias de envío de email marketing, consiste en automatizar la programación de los formularios que se usan en dichas campañas de una forma intuitiva.



Figura 12: Ejemplo de una campaña publicitaria.

Cualquier persona podrá usar la aplicación, dado que no es necesario tener conocimiento de programación web, solo necesitará registrarse en la aplicación con el correo personal de la empresa, elegir un nombre de usuario y una contraseña, y empezar a usar la aplicación.

Un ejemplo básico sería: Un usuario inicia sesión en la aplicación, sube un formulario web y paso a paso va realizando el proceso de validación mediante una serie de campos desplegables que irá seleccionando.

Cuando finalice la validación del formulario, podrá comprobar que este funciona correctamente y posteriormente podrá descargarlo con las validaciones incluidas mediante una llamada al fichero JavaScript que se encontrará en el servidor.

Al entrar en la aplicación el usuario podrá seleccionar entre subir un formulario nuevo para validar, visualizar los posibles formularios que tenga en su directorio, configurar su perfil, acceder al manual de usuario y salir de la aplicación.

La interfaz de usuario de la aplicación debe ser amigable e intuitiva, esto es importante para evitar conflictos con los usuarios finales y de esta forma fomentar su uso en la empresa.

Con esto se pretende que la creación y validación del formulario se realice en un mismo paso y de esta forma no sea necesario pasar por el departamento de IT, por lo que se optimizaría el tiempo en la creación de una campaña de email marketing.

3.3. Objetivos.

Los objetivos de este proyecto son:

- Realizar un análisis comparativo de las tecnologías y soluciones similares existentes.
- Crear una interfaz de usuario amigable con soporte a la inserción de plantillas, creación, programación y borrado de formularios.
- Dar soporte de validación dependiente del país donde se realice la campaña publicitaria.
- Analizador de plantillas con generación automática de validación de formularios.
- Despliegue automatizado en servidor de las plantillas creadas.

3.4. Fases del desarrollo.

El proyecto se desarrolló a lo largo del primer semestre del curso 2014-2015, y se realizaron aproximadamente 325 horas. Para ello, se creó un plan de trabajo y se dividió en 7 fases:

Fase 1. Investigación y Análisis

Esta fase consiste en la búsqueda de información y análisis de los posibles lenguajes de programación y frameworks¹² que se podrán usar en el proyecto. Se analizará las ventajas e inconvenientes de cada uno y se realizarán ejemplos básicos para ver su funcionamiento. Una vez analizada la información buscada, se realizará una inmersión, para elegir la que mejor se adapte a nuestras necesidades.

Fase 2. Preparación del entorno de trabajo.

El objetivo de esta tarea es instalar y probar el entorno de desarrollo, para luego realizar pruebas básicas. Esto es útil para realizar un acercamiento inicial con respecto al desarrollo del proyecto.

Fase 3. Definir los requisitos y etiquetas que se usaran en el formulario web.

En esta tarea se definen los requisitos iniciales de la aplicación junto con el tutor profesional de la empresa. También junto con el departamento creativo se definen las etiquetas que se van a usar dentro de la plantilla del formulario en HTML5.

¹² **Frameworks**, término inglés. Es un marco de trabajo.

Fase 4. Creación de la aplicación Web

- **Diseño MVC**

Una de las fases más importantes del desarrollo es la creación de la arquitectura del programa, en la cual se hará una estructura básica de nuestra aplicación.

- **Implementación**

Después de saber cuáles serán los requisitos y el diseño de la aplicación, se tendrá que implementar el sistema cumpliendo con las especificaciones definidas, siendo esta tarea la más larga.

- **Pruebas**

Una vez terminado el proyecto se realizarán pruebas con usuarios reales, pudiendo detectar los posibles errores y corregirlos.

Fase 5. Instalación y configuración

Se instalará y configurará la aplicación en los sistemas de la empresa Canalmail S.L, estos son: el sistema de pre-producción y el sistema de producción.

Fase 6. Generación de la documentación

Esta fase se realizará a lo largo del semestre, y consiste en redactar varios documentos, tales como: los informes de reunión, el plan de trabajo, el estado del arte, la memoria de seguimiento, la memoria final, la presentación y el manual de usuario.

3.5. Dificultades encontradas.

Una de las dificultades del proyecto fue realizar la taxonomía de los campos, para la validación del formulario y el almacenamiento en la base de datos.

Otra dificultad añadida a este proyecto se debe al país de lanzamiento de las campañas publicitarias, esto se tiene en cuenta cuando se realice el proceso de validación, porque no es lo mismo validar un teléfono español que un teléfono mexicano.

Pero la mayor dificultad que se ha encontrado es el tiempo, dado que en la empresa hay muchas tareas diarias que hacer como por ejemplo resolver incidencias a nivel de usuario y nivel de plataforma, apoyar al desarrollo de otros proyectos informáticos de la empresa, monitorizar el volumen de envío y del rendimiento de las campañas de email-marketing, entre otras.

Por esto se ha tenido que gestionar el tiempo de trabajo entre el aprendizaje de las tecnologías web y el manejo del framework con el que se desarrolla esta aplicación.

3.6. Estructura de la memoria final.

Esta memoria se ha estructurado de la siguiente forma para facilitar su lectura:

- **Entorno:**

Aquí se explica el lugar de trabajo, para quién y el por qué se realiza el desarrollo de este proyecto.

- **Resumen:**

En el que se explica qué es un formulario web, el lugar en el que se desarrolla, y una breve introducción del proyecto para favorecer la comprensión del tema facilitando su entendimiento.

- **Introducción:**

Donde se describe la razón por la que se realizó dicho proyecto, una descripción del trabajo realizado, los objetivos, las dificultades, las fases del desarrollo y la estructura del documento.

- **Estado del arte:**

En donde se expone el análisis de los lenguajes de programación web, librerías, algoritmos y frameworks que se podrían usar para el desarrollo de esta aplicación y que cumplan con los requisitos del sistema.

- **Desarrollo:**

En esta parte se empieza a hablar primero de los requisitos de la aplicación, y se hace un análisis del sistema en el cual se describen las diferentes partes del proyecto. Después, se explica en detalle cada parte de su funcionamiento.

- **Pruebas:**

En dicho apartado se realizan diferentes pruebas con usuarios finales y se analizan los resultados obtenidos.

- **Conclusiones:**

En este apartado se exponen las conclusiones del trabajo realizado.

- **Futuras líneas de desarrollo:**

Aquí se describen las posibles futuras tareas que se podrían realizar para la mejora y continuidad de la aplicación.

- **Bibliografía:**

En este apartado se citan las diferentes fuentes que se han usado para la elaboración de este proyecto.

- **Anexo:**

Contiene la siguiente información: manual de usuario, estructura de la base de datos y los tipos de las validaciones

4. ESTADO DEL ARTE

4.1. Frameworks.

Para organizar la estructura del proyecto siguiendo la arquitectura Modelo Vista Controlador (MVC) se ha decidido usar un framework que implemente esta arquitectura.

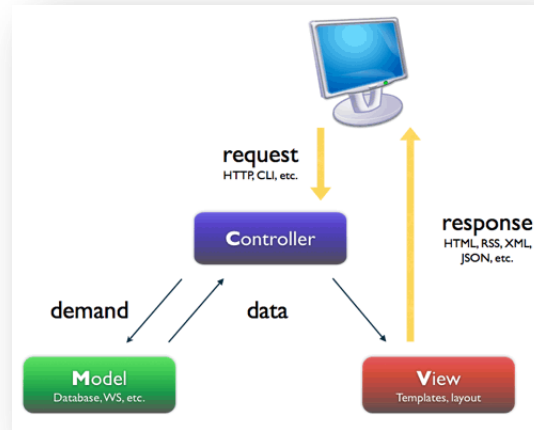


Figura 13: Arquitectura Modelo Vista Controlador. [9]

De esta forma se seguirá una metodología de trabajo organizada. Se han investigado dos frameworks que cumplen este requisito que son: Symfony2 y Zend Framework.

4.1.1. Symfony2 [10].



Figura 14: Logotipo del Framework Symfony. [11]

Symfony2 es un framework de PHP que utiliza la arquitectura MVC. Ha sido construido con componentes independientes que han sido creados por el proyecto Symfony.

Se usa bajo la licencia MIT, es decir es un software libre con el que se puede crear aplicaciones web de una forma rápida y profesional.

Existe una gran comunidad de programadores en todo el mundo que participan activamente en el desarrollo de este framework, esto garantiza la continuidad de los proyectos.

Grandes proyectos como Drupal 8 han sido construidos usando componentes de Symfony.

Pero lo más importante de este framework es la documentación existente en la web en español, lo que ayuda al rápido aprendizaje y a resolver los problemas encontrados de una forma sencilla e eficaz.

4.1.2. Zend Framework. [12]



Figura 15: Logotipo del Framework Zend. [13]

Zend Framework (ZF) es un framework de código abierto para desarrollar aplicaciones Web y servicios Web con PHP 5.

ZF implementa el modelo MVC, es orientado a objetos, y tiene un estándar de codificación que seguir en el proyecto.

Tiene un gran soporte para las aplicaciones con multi-idioma, convertir formatos de fechas, monedas, etc.

Algo muy importante es que es adaptable para distintos tipos de base de datos y además posee componentes para la autenticación y autorización de usuarios ya testeados, por lo que ahorra una gran parte de trabajo.

Dicho framework posee gran documentación y manuales en la web, pero el idioma en el que se encuentra la mayoría es inglés.

4.1.3. Comparación.

Se puede observar en la siguiente imagen una comparación del volumen de búsqueda que hay en internet.

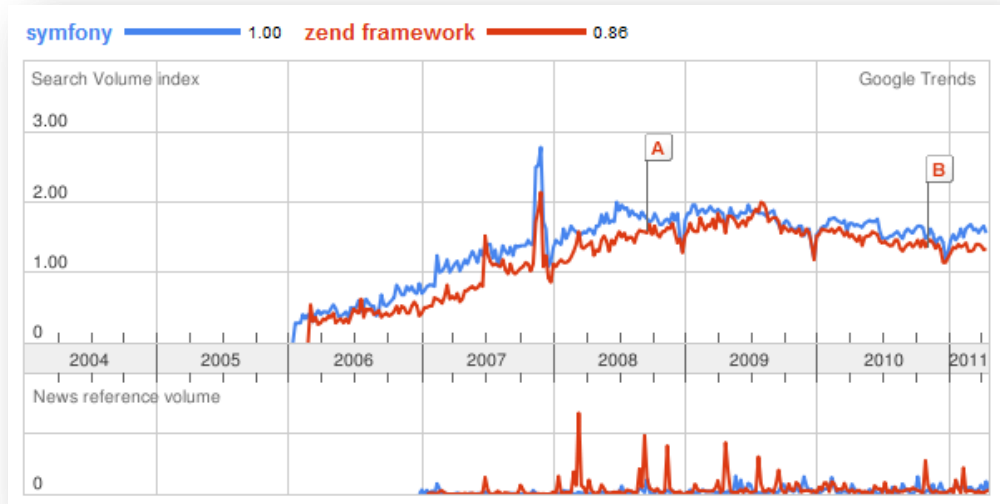


Figura 16: Volumen de búsqueda Symfony vs Zend. [14]

Como se puede observar ambos framework están a la par, y esto se debe a las pocas diferencias que tienen. Sin embargo Symfony cuenta con mayor cantidad y calidad de documentación que Zend Framework.

También hay que destacar que Symfony está mejor orientado al modelo MVC que Zend Framework, dado que este último estaba orientado más a una librería de componentes que al modelo MVC, por lo que dispone de pocos plugins en la comunidad.

En ambos framework el IDE¹⁵ ideal para trabajar es Netbeans [15]. Esta plataforma es un producto libre, gratuito y sin restricciones de uso, que permite crear aplicaciones Web con PHP 5.



Figura 17: Logotipo del IDE Netbeans. [16]

Hay que destacar que este IDE tiene soporte personalizado para Symfony2 y ZF.

¹⁵ **IDE**, en inglés *Integrated Development Enviroment*. Es un entorno de desarrollo de programación.

4.1.4. Decisión final.

Después de analizar los resultados, se observa que los dos frameworks cumplen con los requisitos necesarios para el desarrollo de este proyecto. Por esta razón se han creado dos pruebas con ambos.

Se instaló y configuró los dos frameworks para hacer pruebas y entender su funcionamiento.

Después de ello se ha decidido usar Symfony, en concreto la versión Symfony2, por su gran documentación y material didáctico en español, así como su fácil instalación, aprendizaje y por su gran comunidad en internet que te garantiza su evolución continúa y que no se quedara obsoleto en poco tiempo.

Además, al tener tiempo limitado, este Framework tiene una curva de aprendizaje menor que ZF, ya que ofrece diversos cursos y tutoriales simples para empezar.

4.2. Tecnologías actuales

Este apartado consiste en una búsqueda y un análisis de las tecnologías que mejor se adaptan a las necesidades del proyecto y que puedan cumplir con alguno de nuestros requisitos.

4.2.1. HTML5 [17].



Figura 18: Logotipo de HTML5 [17].

Lenguaje de programación web, es una evolución lógica del lenguaje básico del WWW¹⁸ que surge de la necesidad de conseguir nuevos objetivos, como por ejemplo eliminar las restricciones del código, haciendo que sea más fácil de usar y sobre todo que sea compatible con todos los dispositivos actuales.

Se ha decidido usar HTML5 con respecto a su versión anterior por tener un código más sencillo, nuevos elementos y atributos que ayudan a cargar una página en el navegador web de forma más rápida.

Hay que destacar que HTML5 tiene una mayor compatibilidad al cargar una página en diferentes navegadores web que HTML4.

¹⁸ **WWW**, en inglés "World Wide Web". Sistema de distribución de documentos accesibles vía Internet.

4.2.2. JavaScript [18].



Figura 19: Logo de JavaScript. [19]

Es un lenguaje de programación interpretado, orientado a objetos que se puede modificar en cualquier editor de textos y se ejecuta en un navegador web en el lado del cliente. Su sintaxis es similar al lenguaje de programación C con matices del lenguaje de programación Java, pero no están relacionados dado que tienen propósitos diferentes.

Se ha decidido usar JavaScript por su gran nivel de complejidad y prestaciones que junto con HTML5 hacen posible todo tipo de acciones e interacciones en un sitio web, desde validar formularios hasta realizar la comunicación asíncrona con el servidor por medio de Ajax.

Su interacción con la librería DOM¹⁹ proporciona un conjunto estándar de objetos para representar documentos HTML.

Las librerías JavaScript que se han analizado son: jquery y mootools.



Figura 20: Logotipo de jQuery. [20]



Figura 21: Logotipo de Mootools. [21]

¹⁹ **DOM**, en inglés "Document Object Model". Es una interfaz de programación.

La librería jQuery [22] es un complemento esencial de la web, ayuda a reducir el código JavaScript y en este trabajo es importante sus funcionalidades de programación por medio de plugins para validar formularios, sistemas de plantillas, etc.

Otra librería similar que destaca es “My object oriented tools” [23], conocida como Mootools, esta API es modular y extensible. Además cuenta con una gran documentación en internet.

Una de sus principales ventajas es desarrollar JavaScript sin importar en qué navegador se ejecute.

En la siguiente imagen se puede observar que jQuery es actualmente la API con más resultados de búsqueda en internet, además que empresas importantes como Amazon y Facebook lo usan.

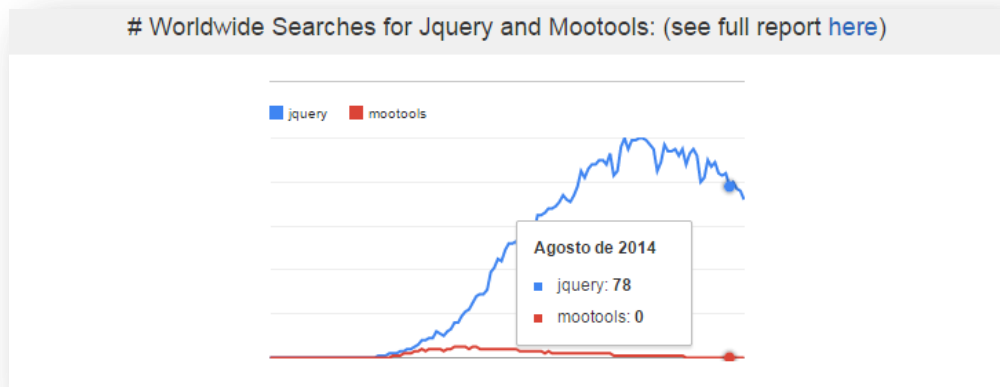


Figura 22: Comparación de búsqueda entre jQuery y Mootools. [24]

Como se observa en la siguiente figura, se analizan los resultados de las APIs:

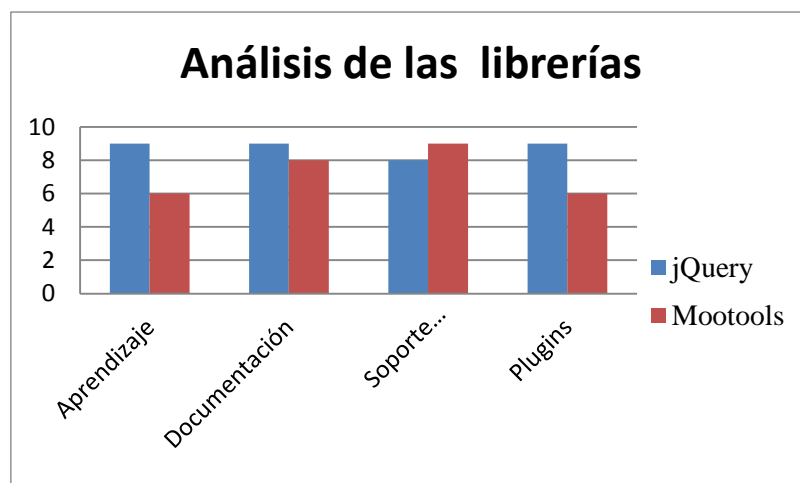


Figura 23: Diferencia entre jQuery y Mootools. [25]

Se puede destacar que la mejor API es jQuery, dada su facilidad de aprendizaje y documentación extensa de sus prestaciones y sus plugins para el desarrollo de aplicaciones web.

Por esto se ha decidido usar la librería jQuery por ser la más adecuada para el desarrollo de este proyecto.

4.2.3. Framework CSS.

Se realizó una búsqueda de un framework CSS para la creación del diseño del front-end de la aplicación.

Después de analizar los resultados de la búsqueda, los framework que más destacan son: Twitter Bootstrap y Zurb Foundation.



Figura 24: Logotipo de Bootstrap. [26]

Bootstrap [27] es un framework creado por Twitter que permite crear interfaces web con CSS y JavaScript. Ayuda a adaptar la interfaz del usuario dependiendo del dispositivo donde se visualice el diseño, a esto se denomina Responsive Design.



Figura 25: Logotipo de Zurb Foundation. [28]

Zurb Foundation [29] es un framework creado por ZURB que dispone de una gran colección de herramientas para la creación de sitios y aplicaciones web.

Ambos frameworks [30] disponen de una documentación amplia y numerosos ejemplos de plantillas con diseño responsivo, además los dos se pueden utilizar actualmente bajo la licencia MIT y cumplen los requisitos necesarios para realizar este proyecto.

Sin embargo se ha decidido utilizar Bootstrap por su presencia en Internet, existe una mayor comunidad en internet de Bootstrap que de Zurb Foundation, esto se debe a que Bootstrap siempre tuvo su código abierto, mientras que Foundation liberó su código en 2011.

4.2.4. PHP [31].

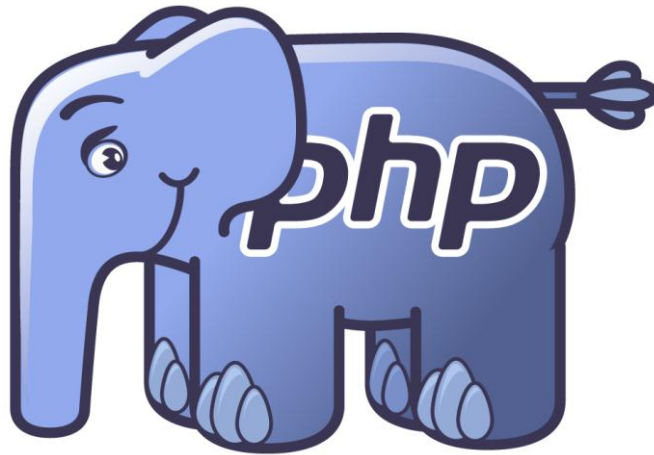


Figura 26: Logotipo de PHP. [32]

El lenguaje de programación web *Hipertext Preprocessor* conocido como PHP, permite interactuar con los usuarios y utilizar sistemas de Bases de Datos. De esta forma se generan páginas web dinámicas con una tecnología avanzada como son HTML5 Y JavaScript que incluyen nuevos elementos para la creación de páginas web.



Figura 27: Icono de Java Server Pages. [33]

Antes de la realización de este proyecto en la empresa usaba la tecnología Java Server Page (JSP²⁰) para la creación de páginas web dinámicas basadas en HTML y XML.

Como ha sido mencionado anteriormente en el apartado de antecedentes del proyecto, este sistema está montado mediante Struts que corre sobre un servidor Tomcat. El mayor problema que se observó fue el rendimiento en la creación de las campañas publicitarias y cabe destacar que al tratarse de una empresa de envío de email marketing esto supone pérdidas.

²⁰ **JSP**, en inglés “Java Server Pages”, es un lenguaje programación web.

Como se observa en la siguiente imagen, existe una diferencia notable entre el sistema actual y el antiguo, PHP y JSP respectivamente. De los distintos parámetros que se evalúan, los más importantes para este proyecto son: la integración y el rendimiento en el sistema y sobre todo el tiempo que conlleva el aprendizaje del lenguaje.

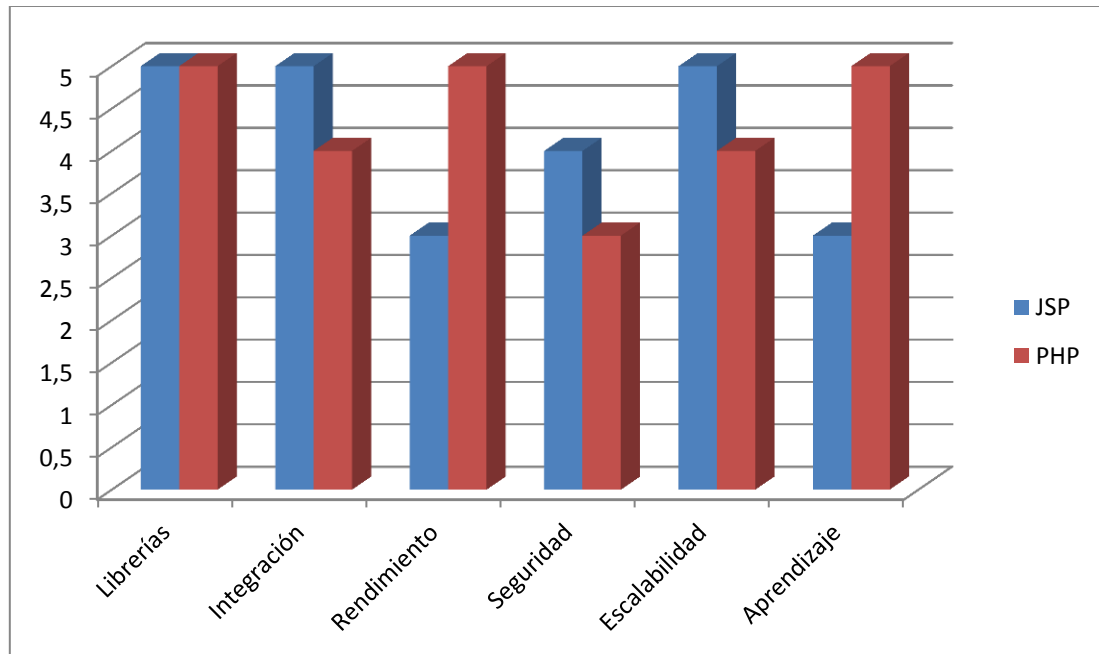


Figura 28: Diferencias entre PHP y JSP. [34]

Dado que el framework que se usa es Symfony2, PHP es el lenguaje más adecuado para programar los servicios que tendrá la aplicación, además de tener las siguientes características: fácil aprendizaje, librerías útiles y un alto rendimiento de sus servicios.

4.2.5. Python [35].



Figura 29: Logotipo del lenguaje de programación Python. [36]

Python es un lenguaje de programación que permite crear diferentes programas incluyendo sitios web.

Una de sus ventajas principales es que su código no necesita ser compilado, es un lenguaje de programación *multiparadigma*²¹, y dispone de una gran cantidad de funciones, librerías, y además tiene licencia Opensource.



Figura 30: Logotipo de Perl. [37]

Aunque es comparado con el lenguaje de programación Perl, el aspecto más evidente está elegir entre la claridad o flexibilidad. Python obliga a cumplir convenciones de buen programador en el código como por ejemplo la *indentación*²², por lo que es considerado un lenguaje limpio, mientras que Perl [38] usa convenciones más tradicionales como por ejemplo el uso de llaves para delimitar las funciones.

Para este trabajo se ha decidido usar Python, en vez de Perl, por la familiaridad ya existente con el lenguaje en el desarrollo de otros programas para la empresa y la afinidad con el lenguaje PHP y el framework Symfony2.

²¹ **Multiparadigma**, es un conjunto de estilos de programación orientada a objetos.

²² **Indentación**, mover un bloque de texto hacia la derecha insertando espacios o tabuladores.

4.3. Algoritmos.

Se han analizado diferentes algoritmos de búsqueda para encontrar los diferentes campos que se quieren validar en un determinado formulario y sustituir sus atributos.

4.3.1. Algoritmo de búsqueda: Boyer-Moore. [39]

Este algoritmo fue desarrollado por Bob Boyer y Strother Moore en el año 1977. Este algoritmo pre-procesa la cadena clave que se busca, pero no en la cadena en que se busca.

Características:

- Realiza las comparaciones de derecha a izquierda
- La fase de pre-procesamiento se realiza en $O(m+\sigma)$ en tiempo y espacio.
- La fase de búsqueda toma $O(mn)$ en tiempo.
- En el peor caso se necesitan complejidad $3n$ comparaciones de caracteres cuando se busca un patrón no periódico.
- En el mejor caso toma $O(n/m)$.

Un ejemplo básico es:

f a b f a a b b d a b d a a b a b d a a b	Algoritmo1=1 Algoritmo2=0 Salto=1
f a b f a a b b d a b d a a b a b d a a b	Algoritmo1=3 Algoritmo2=4 Salto=4
f a b f a a b b d a b d a a b a b d a a b	Algoritmo1=1 Algoritmo2=4 Salto=4
f a b f a a b b d a b d a a b a b d a a b	Hay completa concordancia. Fin del proceso. Cadena encontrada.

Figura 31: Ejemplo de aplicación del algoritmo Boyer-Moore. [40]

4.3.2. Algoritmo Knuth-Morris-Pratt [41].

También conocido como **KMP** es un algoritmo que busca la existencia de una sub-cadena dentro de una cadena.

Consiste en localizar la posición de comienzo de una cadena, dentro de otra y calcula una tabla de saltos conocida como tabla de fallos, que se examina para localizar la cadena buscada.

Un ejemplo de aplicación de este algoritmo con una sola ocurrencia es:

0	1	2	3	4	5	6	7
T	A	N	G	E	N	T	E
-1	0	0	0	0	0	0	1

Tabla 1: Ejemplo de la aplicación del algoritmo KMP

Explicación del ejemplo:

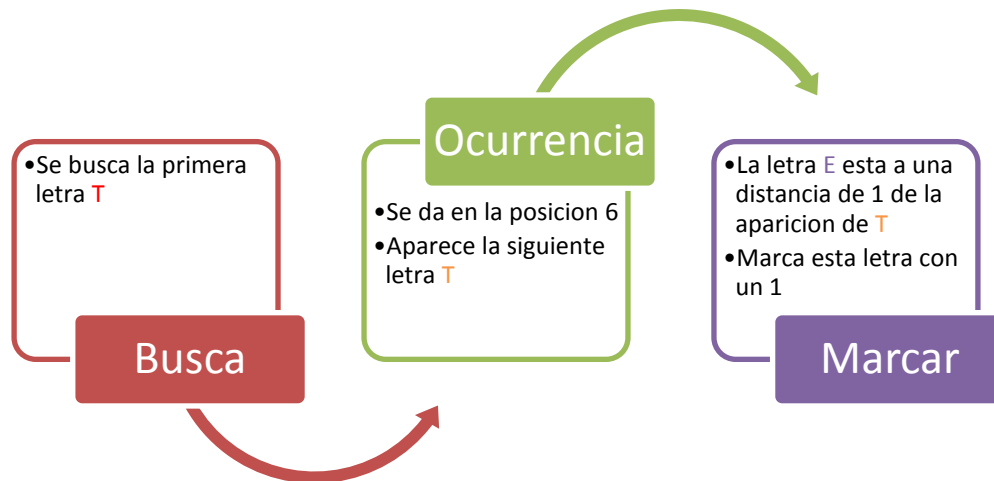


Figura 32: Aplicación del algoritmo KMP.

4.3.3. Decisión del algoritmo.

Ambos algoritmos tienen implementación en el lenguaje de programación Python.

Para el desarrollo de los módulos del analizador y modificador del proyecto, se ha usado una metodología similar que en el algoritmo KMP, por su fácil comprensión.

La búsqueda de una sub-cadena dentro de una cadena, en el proyecto se aplicará para la búsqueda de los campos a validar dentro del cuerpo del formulario, usando unas etiquetas preestablecidas anteriormente.

4.4. Base de datos.

MySQL [42] es un sistema de gestión de bases de datos, una de las más usadas en el mundo por ser un software libre y además potente, rápido, seguro y fácil de usar.



Figura 33: Logo de MySQL.

Existen más sistemas libres como PostgreSQL [43] o SQLite [44], y otros de pago como Progress [45] o Informix [46].

El inconveniente de usar Progress o Informix es la falta de ayuda técnica, y por otra parte, al tener una pequeña comunidad, es más difícil encontrar soluciones en internet.

El problema de usar PostgreSQL es la cantidad de recursos que consume, y es de dos a tres veces más lento que MySQL.

5. DESARROLLO

5.1. Requisitos.

Antes de empezar a implementar la aplicación, se establecieron una serie de requisitos para especificar su alcance.

En este apartado se definen las funcionalidades mínimas que tendrá la aplicación.

5.1.1. Lista de requisitos.

La aplicación tiene las siguientes funcionalidades:

1. Crear cuenta de usuario:

- Req 1.1 Registro del usuario, formulario de alta de usuario
- Req 1.2 Formulario de alta de usuario tiene los siguientes campos: nombre de usuario que deberá ser único, email de usuario y contraseña cifrada md5 por seguridad.
- Req 1.3 Se redirige al formulario de acceso.

2. Conexión a la aplicación

- Req 2.1 El formulario de acceso tendrá dos campos de texto: nombre de usuario y contraseña. A demás tendrá un botón de enlace de registro.
- Req 2.2 Al identificarse, se redirige a la perfil del usuario

3. Perfil de usuario

- Req 3.1 El menú de usuario contiene las siguientes páginas: Subir formulario, Mis formularios, Configuración, Ayuda, Salir.
- Req 3.2 Al hacer clic en salir, el usuario se desconecta y vuelve al formulario de acceso
- Req 3.3 Al hacer clic en subir formulario redirige a cargar formulario.

4. Cargar formulario

- Req 4.1 Aquí el usuario selecciona un fichero cuya extensión tiene que ser **‘.html’** si es otro tipo de extensión devuelve un error.
- Req 4.2 Se carga el formulario, se visualiza el formulario tal cual para asegurarse de que el diseño es el correcto.
- Req 4.3 Se elige el país del lanzamiento de la campaña para usar las validaciones propias de dicho país.

5. Validación de los campos

- Req 5.1 Se cargan los campos que se van a validar.
- Req 5.2 Se muestra una lista con todas las posibles validaciones que puede tener un campo.

- Req 5.3 Se muestra una lista con los campos de la base de datos en donde se guardan los datos del Req 5.2
- Req 5.4 Una vez validado se modifica el formulario original para asignar a cada campo su validación y el campo en el que se guardará dicho dato.
- Req 5.5 Una vez modificado, se podrá descargar el formulario.
- Req 5.6 También se podrán realizar pruebas en la aplicación para saber si es correcta o no la validación de los campos.

6. Mis formulario

- Req 6.1.El usuario podrá listar todos los formularios de su directorio.
- Req 6.2 El usuario podrá visualizar cualquier formulario de su directorio.
- Req 6.3 El usuario podrá borrar cualquier formulario de su directorio.

7. Configuración

- Req 7.1 El usuario podrá cambiar su contraseña.

8. Ayuda

- Req 8.1 Se despliega el manual del usuario.

5.2. El sistema.

En este apartado se explicarán las diferentes partes que interactúan en el proyecto y su funcionamiento para tener una idea general del sistema.

5.2.1. Componentes del proyecto.

En este proyecto existen 4 partes importantes que forman el proyecto, de los que se hablará en detalle en el siguiente apartado.

- **Sistema de usuarios**

Para empezar se tuvo que crear la estructura que tendrá el sistema de usuarios, es decir cómo se realizará el proceso de registro y el inicio de sesión de los usuarios, qué parámetros son necesarios para llevar a cabo este procedimiento y para la buena gestión de la aplicación.

- **Función principal: Programar un formulario.**

- **Módulo analizador**

Este componente es muy importante para realizar correctamente las validaciones de los campos del formulario. Este módulo se encarga de leer el formulario original que el usuario sube en la aplicación y detectar que campos se tienen que validar.

- **Validaciones**

Esta es una de las partes más importantes del proyecto dada la complejidad que tiene, aquí se realizará un despliegue de la clasificación de todos los campos a validar que tiene un formulario dependiendo de los requisitos de las campañas publicitarias.

- **Módulo modificador**

Este módulo se encarga de copiar en el formulario original los campos que se han validado y además añade la ruta necesaria del fichero JavaScript que contiene dichas validaciones

- **Sistema de gestión de base de datos**

Existen dos sistemas de gestión de bases de datos que se utilizan en este proyecto que son: MySQL y Oracle.

En la base de datos de MySQL se guardan los datos del usuario y los datos de las pruebas que se realizan de los formularios validados.

En la base de datos Oracle se encuentran los sistemas reales de producción de la empresa.

Se tendrán que enviar los datos de la base de datos Mysql local a la base de datos central de la empresa que es Oracle.

- **Gestión de formularios.**

Para que el usuario encuentre la aplicación accesible, en el perfil de usuario encontrará un menú con los siguientes apartados: Subir Formulario, Mis formularios, Configuración, Ayuda y Salir.

Con estos apartados el usuario podrá gestionar de una forma sencilla el uso de la aplicación que se explicará en detalle más adelante.

En el apartado *Mis Formularios* el usuario podrá gestionar su directorio, que contendrá todos los formularios que ha programado.

5.3. Sistema de usuarios.

5.3.1. Usuarios finales

Este proyecto está destinado para el departamento creativo del corporativo Media Response de España, en la siguiente imagen se presenta al personal.



Figura 34: Departamento creativo de la empresa. [47]

Estos son los trabajadores del departamento creativo, que serán los usuarios finales de esta aplicación Web dado que son los encargados de maquetar los formularios online siguiendo los requisitos que dan los clientes dentro de cada campaña publicitaria.

Antes de empezar el desarrollo del proyecto se hicieron varias reuniones con todo el departamento creativo y el tutor profesional de la empresa para especificar las necesidades que tiene esta aplicación Web.

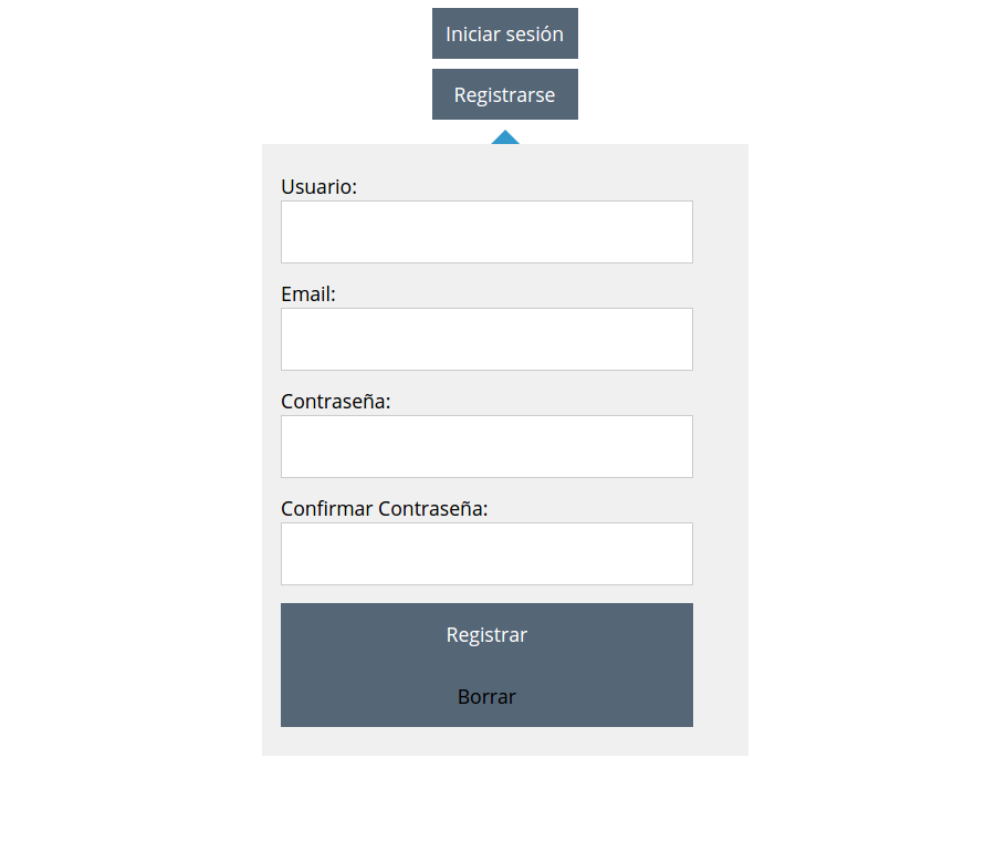
En la primera reunión se les explicó la idea general del proyecto, obteniendo valoraciones positivas.

En las siguientes reuniones se establecieron una serie de etiquetas que se van a usar dentro de cada plantilla de formulario, estas etiquetas servirán como identificadores para el proceso de validación de la aplicación (Se explican más adelante).

Se ha escrito un manual de usuario (ver Anexo), en el que se explica detalladamente como realizar la programación del formulario, qué campos hay que usar y qué tipo de validaciones existen dependiendo de la campaña publicitaria a la que pertenece el formulario web.

5.3.2. Datos necesarios para acceder al sistema.

Para usar la aplicación web es necesario registrarse. Para esto es necesario rellenar los siguientes parámetros obligatorios: nombre de usuario, email y contraseña



The screenshot shows a registration interface. At the top, there are two buttons: 'Iniciar sesión' and 'Registrarse'. Below these, a form is displayed with the following fields and buttons:

- Usuario: [input field]
- Email: [input field]
- Contraseña: [input field]
- Confirmar Contraseña: [input field]
- Registrar [button]
- Borrar [button]

Figura 35: Pantalla de registro.

1) Nombre de usuario

El nombre de usuario es el nombre único que tiene cada trabajador dentro de la empresa, esta aplicación tiene conexión con el *sistema de Auth*.

Este sistema Auth, es un proyecto interno de la empresa que tiene una base de datos con los datos personales de cada trabajador de la empresa, en este caso solo nos interesa el nombre de usuario para realizar la autenticación. Esto es necesario por seguridad. De esta forma se asegura que nadie externo a la empresa pueda acceder a esta aplicación Web.

2) Email

El correo electrónico que se usa para realizar el registro, es el personal de cada trabajador de la empresa. Al igual que con el nombre de usuario se usa el sistema Auth con el correo electrónico por seguridad.

3) Contraseña

La contraseña por motivos de seguridad esta codificada con MD5 y tiene una longitud mínima de 8 caracteres que debe contener letras y números.

MD5 son las siglas de “Message Digest Algorithm”. Es un algoritmo de reducción criptográfico de 128 bits, que fue diseñado por el profesor Ronald Rivest del Instituto Tecnológico de Massachussetts en 1991 como una mejora del algoritmo MD4. Es comúnmente usado para codificar contraseñas en bases de datos.

Estos datos se guardan en la tabla de usuarios de la base de datos del proyecto usando el sistema de gestión de base de datos Mysql y la conexión con esta base de datos mediante Symfony2 se realiza usando un fichero de configuración.

Cuando el usuario se ha registrado recibirá un correo electrónico de bienvenida, recordándole que para acceder a la aplicación web deben utilizar su nombre de usuario y contraseña.

El diagrama muestra la interfaz de usuario para iniciar sesión. En la parte superior, hay dos botones rectangulares: 'Registrarse' y 'Login'. Debajo de ellos, un botón más grande y ancho con el texto 'Iniciar Sesión'. En el centro, hay dos campos de entrada de texto: el primero está etiquetado como 'usuario' y el segundo como 'Password'. En la parte inferior, hay un botón rectangular con el texto 'login'.

Figura 36: Pantalla para iniciar sesión.

Al inicio de sesión el usuario podrá acceder al menú de la aplicación y empezar a programar los formularios.

5.4. Función principal: Programación de un formulario.

Antes de entrar en detalle de cada parte, se explicará un resumen sencillo para entender mejor el funcionamiento general de la aplicación.

Para que el usuario pueda programar un formulario, sigue los siguientes pasos:

- Carga y visualización del formulario.
- Módulo analizador.
- Selección de las validaciones y de los campos de la BBDD.
- Módulo modificador.
- Finalización.

5.4.1. Carga y visualización del formulario.

Para empezar, el usuario sube el formulario que se desea programar, y lo visualiza para asegurarse de que las imágenes y los campos se ven correctamente.

5.4.2. Módulo analizador.

En el siguiente paso, se ejecuta el script analizador, que obtiene todos los campos del formulario, siendo los “input”, “select” y “checkbox”.

5.4.3. Selección de las validaciones y BBDD.

Para cada campo del formulario, el usuario podrá seleccionar que tipo de validación se le quiere asignar, y en qué columna de la base de datos se desea guardar el dato.

5.4.4. Módulo modificador.

Posteriormente, se lanza el modulo modificador, que cambia el formulario para introducir las diferentes validación y la lista de columnas de donde se guardaran los datos.

5.4.5. Comprobación del formulario.

Al finalizar la generación del nuevo formulario, el usuario podrá probarlo online y descargarlo.

5.5. Carga y visualización del formulario.

Para facilitar al usuario el uso de la aplicación web, después de iniciar sesión podrá visualizar el menú principal de la aplicación.



Figura 37: Menú al acceder al panel.

Dentro del proyecto existe un directorio denominado *Users*, en el que se crea un directorio privado por cada usuario que se registre en el sistema.

Es importante destacar que sólo puede acceder al contenido del directorio, el dueño de este, mientras esté conectado al sistema.

Cualquier usuario que no se haya registrado en la aplicación o no esté conectado, no podrá acceder a la información si la hubiera.

5.5.1. Visualización

Para visualizar el formulario se realizan los siguientes pasos:

i. Subir el formulario

El usuario hace clic en la opción de subir formulario tal y como se muestra en la siguiente imagen:

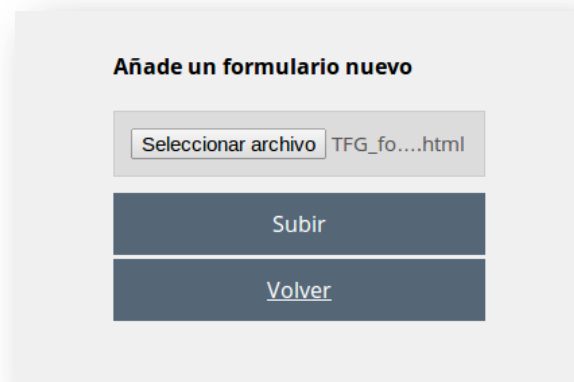


Figura 38: Cargando un nuevo formulario.

Luego podrá seleccionar el formulario web que se va a programar, este deberá tener una extensión **.html**, siendo un requisito del sistema.

Si el usuario sube un formulario válido para la aplicación, puede continuar con el siguiente paso que es la programación del formulario, de lo contrario devolverá un mensaje de error informando que el fichero debe tener la extensión correcta.

ii. Visualizar el formulario en la aplicación

Se visualiza el formulario en la aplicación de la misma forma como si se tratara de un navegador web, esto es importante para observar los errores comunes que se dan al diseñar el formulario web, que son:

- Las imágenes del formulario no se carguen correctamente.
- Los caracteres del formulario no son correctos.
- Diseño no es “responsive”.
- Los botones no se ven bien.
- Los campos no se ven bien.

La mayoría de estos errores se deben a problemas con la localización exacta de un archivo o directorio que necesitan.

Esta localización se da mediante una cadena de caracteres concreta que se denomina ruta o en inglés path.

La causa principal de estos problemas se da en el momento de maquetar los formularios, el departamento creativo encargado de esta tarea, trabaja en su propio directorio por lo que no necesita añadir las rutas completas. Sin embargo una vez que se sube el formulario al servidor de preproducción para continuar con el proceso de programación sí que es necesario añadir las rutas completas.

Es por eso que en la aplicación antes de empezar el proceso de programación de validación de los campos de un formulario es indispensable visualizar el formulario para comprobar que el usuario ve correctamente el diseño del formulario web.

The image shows a web browser window displaying a test application. On the left, there is a logo for the 'Facultad de Informática Univ. Politécnica de Madrid' featuring a shield with a crown and a compass, and the word 'POLITÉCNICA' below it. On the right, there is a form titled 'Aplicación de prueba' with the following fields: 'Nombre' (text input), 'Apellidos' (text input), 'Sexo' (dropdown menu), 'Teléfono' (text input), 'Código Postal' (text input), 'Email' (text input), 'Fecha de nacimiento:' (with 'Día', 'Mes', and 'Año' dropdowns), 'Nacionalidad' (dropdown menu), and 'NIE' (text input). Below these fields is a checkbox labeled 'He leído el Aviso legal' and a red 'Enviar' button. At the bottom of the browser window, there is a dark grey bar with two buttons: 'Correcto' and 'Incorrecto'.

Figura 39: Ejemplo de visualización de un formulario ficticio.

Hay dos opciones:

- **Correcto**
Si el usuario ve correctamente el diseño del formulario, puede continuar con el siguiente paso.
- **Incorrecto**
Si el usuario no ve correctamente el diseño del formulario, se le mostrara una lista con los errores habituales, ya anteriormente mencionados.



Figura 40: Vista de la aplicación mostrando los errores habituales.

El usuario podrá seleccionar todos los errores que aprecie en su formulario y luego se le mostrará un mensaje con la información necesaria para corregir estos errores.

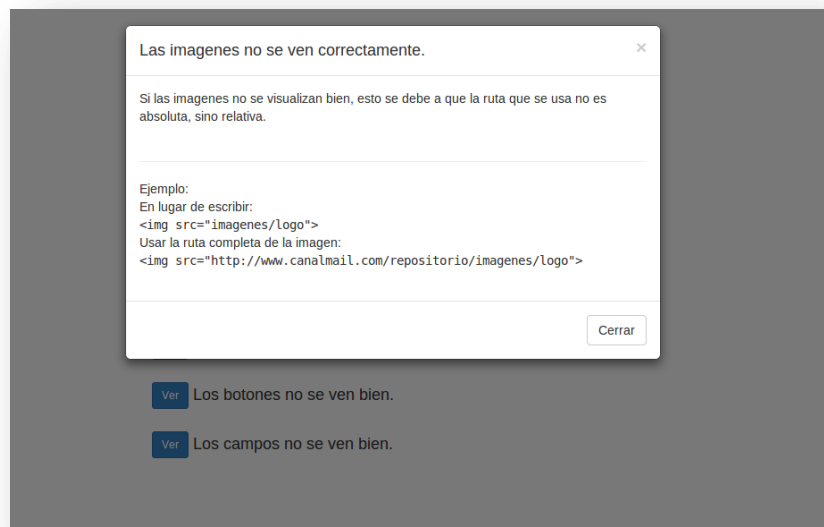


Figura 41: Mensaje con la solución de un error.

5.6. Módulo analizador.

Una de las partes más complejas del proyecto es el analizador, ya que se tiene que ir leyendo dentro del formulario web las etiquetas que se establecieron en las reuniones que se tuvieron con el departamento creativo de la empresa, para detectar los campos que se van a validar.

Para llevar a cabo el analizador del programa se utiliza el lenguaje de Python, por la familiaridad con este lenguaje y el tiempo de ejecución de los ficheros es óptimo.

El objetivo principal de este fichero ejecutable es obtener en un fichero auxiliar todos los campos que se van a programar.

5.6.1. Funcionamiento del script.

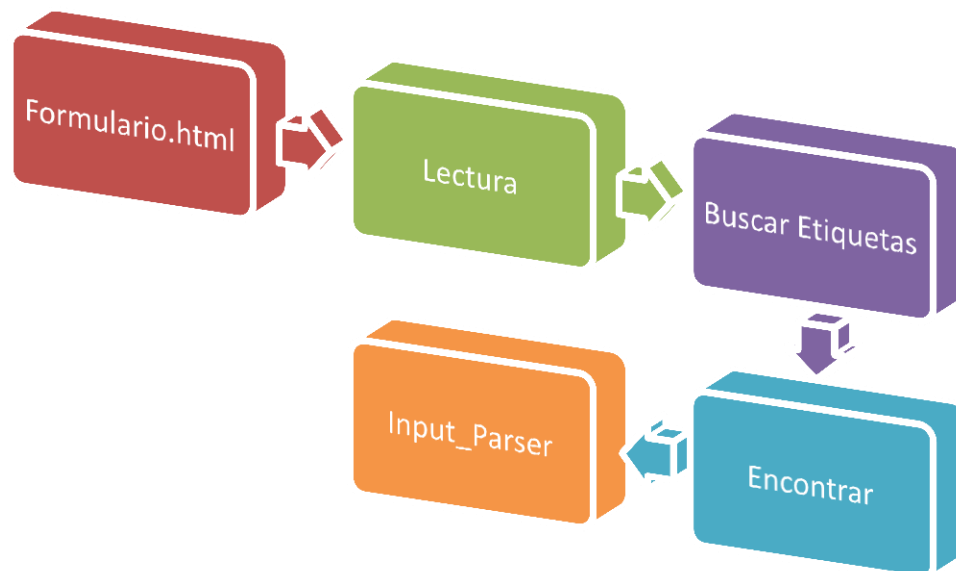


Figura 42: Flujo de trabajo del script Analizador.

5.6.1.1. Entrada: Formulario.html

El script recibe un fichero por la entrada estándar, este fichero es el formulario que se sube en la aplicación para ser validado.

La plantilla del formulario se ha dividido en cuatro bloques:



Figura 43: Segmentación del diseño del formulario.

Cada bloque del formulario tiene dos etiquetas que identifican el inicio y el fin del bloque.

Estas etiquetas se encuentran dentro de la plantilla del formulario como comentarios para no alterar el diseño del formulario.

Estos comentarios en HTML se usan para explicar qué se está haciendo en el código de la página web, pero en este caso va más allá, para este script sirve como punto de referencia para situarse en cada momento mientras se esté leyendo el fichero.

La etiqueta del comentario en HTML tiene la siguiente estructura:

Etiqueta de comentario en HTML5: `<!--comentario-->`

Figura 44: Etiqueta de comentario en HTML5.

Dentro de esta etiqueta, se encuentran las propias de la aplicación, por ejemplo:

Etiqueta de inicio del código HTML5: `<!--I_inicio-->`

Figura 45: Etiqueta de inicio del código HTML5.

5.6.1.2. Lectura

El siguiente paso es abrir el fichero en modo lectura y recorrer línea a línea el formulario en busca de las etiquetas marcadas dentro del código html5 del formulario.

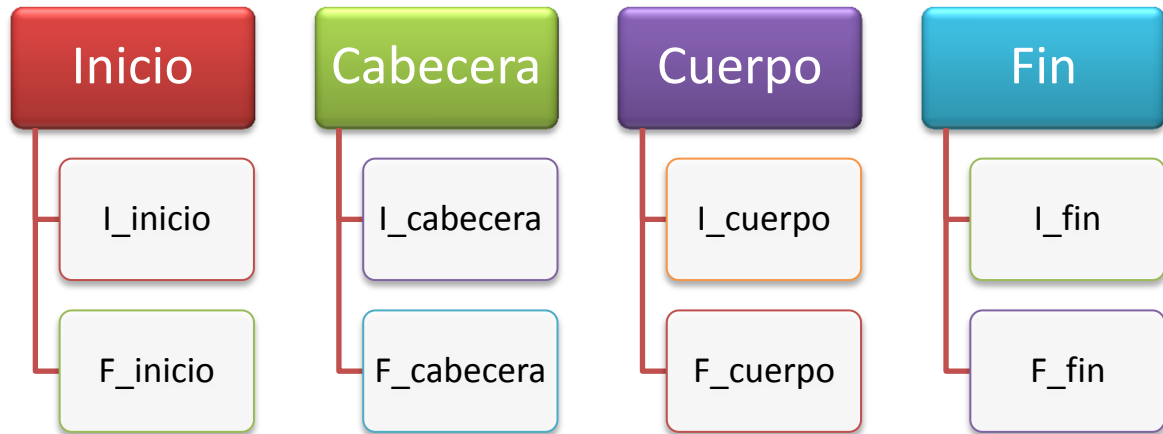


Figura 46: Etiquetas de los principales bloques del formulario.

5.6.1.3. Buscar etiquetas

Dentro del **bloque Cuerpo** del formulario se encuentra toda la información de los campos del formulario con las siguientes etiquetas:

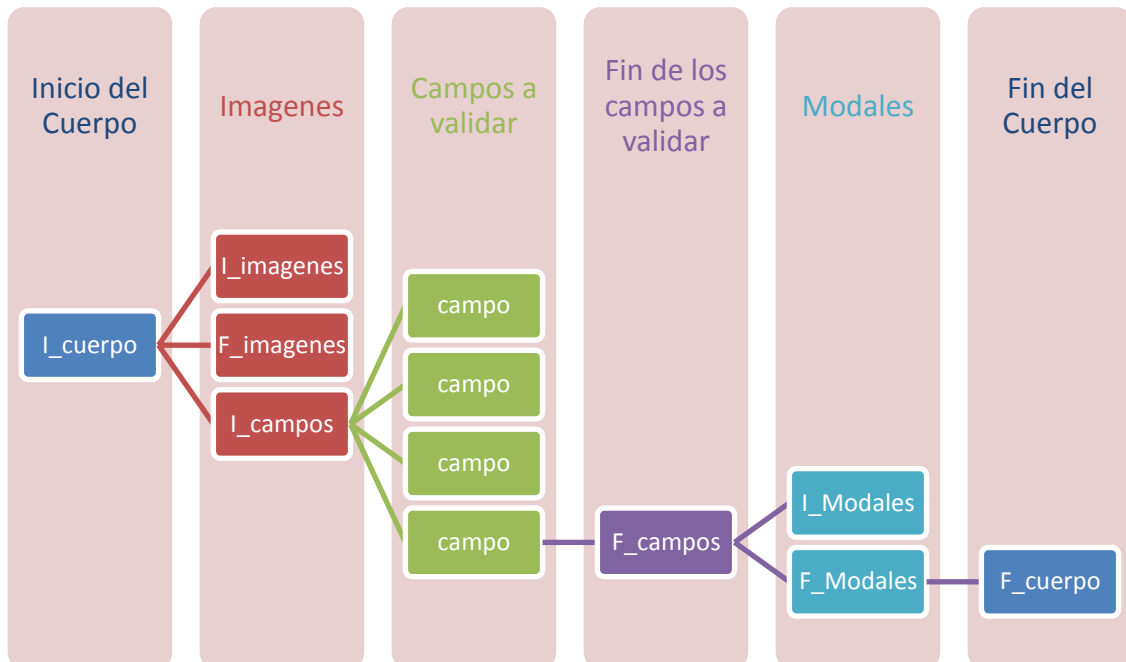


Figura 47: Etiquetas dentro del bloque cuerpo del documento HTML5.

El script **Analizador** lee las etiquetas hasta posicionarse en la etiqueta **I_campos**.

5.6.1.4. Encontrar

Una vez encontrada esta etiqueta el script analiza el formulario buscando todas las etiquetas claves **campo** y a la vez que va encontrando procesa la información quedándose exclusivamente con la información del campo que se va a validar.

Por ejemplo: Dado el siguiente fragmento de código de un formulario en html5

<!--campo-->

```
<div class="form-group">
```

```
<div class="col-xs-2">
```

```
<label class="control-label">Nombre</label>
```

```
</div>
```

```
<input type="text" class="form-control" id="Nombre"
name="campo" placeholder="Nombre" />
```

<!--campo-->

```
<div class="col-xs-2">
```

```
<label class="control-label">Apellidos</label>
```

```
</div>
```

```
<input type="text" class="form-control" id="Apellidos"
name="campo" placeholder="Apellidos" />
```

```
</div>
```

Figura 48: Estructura de un formulario con los campos que se van a procesar.

5.6.1.5. Salida: Input Parser

El script detecta por la etiqueta **campo** que lo que va a leer a continuación es la información necesaria del código del campo que se va a programar.

A continuación se muestra la parte del código necesaria que se copia en un fichero auxiliar llamado **Input_Parser.html** que posteriormente será procesado por la aplicación.

Input_Parser.html

```
<input type="text" class="form-control"
id="Nombre" name="campo"
placeholder="Nombre" />
```

```
<input type="text" class="form-control"
id="Apellidos" name="campo"
placeholder="Apellidos" />
```

Figura 49: Contenido del fichero Input_Parser.html.

5.7. Selección de las validaciones

La programación del formulario depende inicialmente de dos factores: el país al que está destinada la campaña y los campos que se van a validar.

5.7.1. Clasificación.

Este proyecto está destinado como una ayuda al uso comercial de las campañas publicitarias que se lanzan en la empresa, por ello se clasifica las validaciones dependiendo del país de lanzamiento, siendo este el factor más importante.

Las campañas de email marketing se distribuyen en los siguientes países:

- España
- Portugal
- México
- Países de Latinoamérica como: Perú, Argentina, Chile, Colombia, Venezuela y Brasil.

Esto se da por la presencia de la empresa Canalmail S.L en estos países y por su base de datos de más de 40 millones de usuarios con permisos necesarios para enviarles email marketing.

Por ello los principales países que cuentan con una validación propia son España, Portugal y México. Sin embargo, Latinoamérica cuenta con una validación genérica, debido a la moderada presencia de la empresa en estos países

Un usuario que quiera realizar la programación de un formulario, antes de empezar con la validación de cada campo, deberá seleccionar el país al que se destina la campaña, de esta forma en el siguiente paso se cargará una lista con todas las validaciones propias del país.

5.7.2. Lista de validaciones.

Se va a realizar una lista con todas las validaciones existentes para un determinado formulario independiente del país al que esté destinada la campaña publicitaria.

5.7.2.1. *Validación por país de lanzamiento de la campaña*

Esta es la validación más importante, el usuario debe elegir el país de lanzamiento de la campaña publicitaria para cargar el conjunto de validaciones propias de cada país. Es decir, cuando se selecciona el país de la campaña, los campos del formulario tendrán una validación específica de ese país.

5.7.2.2. *Validación por nombre.*

Este tipo de validación solo puede contener letras, si el campo contiene cualquier otra cosa, como números, se muestra un mensaje de error y el usuario deberá corregir el dato que introduce en el campo nombre.

Además la validación por nombre tiene dos opciones: validar un nombre o validar dos nombre o nombre compuesto.

Este caso se da en países como en México que la mayoría de sus habitantes tienen nombres compuestos, en este caso, esta validación acepta dos cadenas de caracteres separadas por un espacio.

5.7.2.3. *Validación por apellido*

La validación por apellido solo puede contener letras sin números y si un usuario introduce cualquier otra cosa mostrará un mensaje de error cuando se pruebe el formulario después de validarlo.

En esta validación se acepta por defecto que el usuario pueda introducir tanto un apellido o los dos apellidos, esta es la diferencia que existe con la validación por nombre.

Por ejemplo: Si la campaña publicitaria se lanza para el territorio español, el anunciante puede pedir como requisito de la campaña que los usuarios que se registren en el formulario introduzcan un solo nombre, pero en países con México donde es más común tener un nombre compuesto, el anunciante en este caso puede aceptar registros con un solo nombre o nombres compuestos. Sin embargo, esto no ocurre con el campo Apellido.

5.7.2.4. *Validación por email.*

Esta validación es común para cualquier país al que este destinado una campaña publicitaria. Aquí se comprueba que el dato introducido cumpla las características de un correo electrónico, esto se realiza mediante una expresión regular que controla que el correo introducido sea correcto.

5.7.2.5. Validación por teléfono.

Este tipo de validación es única para cada país al que está destinada la campaña, esto se debe a que los números telefónicos de la red fija nacional son distintos.

Para entender mejor esta validación, se explicará un ejemplo sencillo:

El teléfono fijo en España consta de 9 dígitos mientras que en México consta de 10 dígitos y el teléfono Móvil en España tiene 9 dígitos pero en México tiene 13 dígitos.

Como se observa en las siguientes tablas las validaciones telefónicas son distintas tanto para teléfonos móviles como para teléfonos fijos. Además en México, para realizar llamas de larga distancia es necesario conocer la clave lada, esto es el indicativo telefónico

País	Lada	Prefijo	Teléfono Fijo
España	-	9	8 dígitos
México	55/01	-	8 dígitos

Tabla 2: Patrón telefónico según el país de teléfonos fijos [48]

País	Lada	Prefijo	Teléfono Móvil
España	-	6/7	8 dígitos
México	044/045	55/01	8 dígitos

Tabla 3: Patrón telefónico según el país de teléfonos móviles [49]

5.7.2.6. Validación por fecha de nacimiento.

Dependiendo del formulario este tipo de validación tiene dos opciones: un campo donde el usuario escriba su edad o un desplegable que seleccione el día/mes/año.

Si es la primera opción, mediante una consulta realizada contra la base de datos se realiza una conversión del dato introducido para obtener el año de nacimiento y almacenarlo en la base de datos.

5.7.2.7. Validación por género

Existen dos tipos diferentes de validación en este caso que son: El usuario selecciona su sexo desde una lista desplegable o mediante un botón de selección.

En cualquier caso el campo género no puede estar vacío y se guarda en la base de datos con el valor **M** o **F**, donde M=Hombre y F= Mujer.

5.7.2.8. Validación Nacionalidad

En esta validación el usuario selecciona de un desplegable el país de nacionalidad.

Este campo tiene codificación ISO 3166-1 alfa-2, esto quiere decir que el código del país es de dos letras. Por ejemplo:

País	Codificación
España	ES
México	MX
Portugal	PT

Tabla 4: Tipo de codificación ISO dependiendo del país. [50]

Se aplica esta codificación para facilitar la integración de los datos con el sistema del anunciante.

5.7.2.9. Validación por dirección

Con este tipo de validación se encuentra la dirección de un domicilio, para el campo dirección hay varios tipos de validación diferentes que son: tipo de calle, número del portal y letra del piso. Esta validación es un conjunto de una cadena de caracteres alfanuméricos.

5.7.2.10. Validación por país de residencia

Esta validación por país de residencia es distinta a la validación por lanzamiento de campaña, ya que el usuario selecciona de una lista desplegable el país en el que reside.

Este campo es importante para las campañas que se lanzan en Latinoamérica, de esta forma se puede analizar en qué país ha tenido más éxito la campaña publicitaria lanzada.

Además este campo también tiene codificación ISO 3166-1 alfa-2.

5.7.2.11. Validación por ciudad

En esta validación se encuentra con dos opciones, el usuario puede rellenar el campo con una cadena de caracteres o puede seleccionar de una lista desplegable.

Por ello existen dos posibles validaciones, la primera que solo admite un conjunto de caracteres no numéricos y la segunda obliga a seleccionar una ciudad de la lista dada.

En ambos casos no puede estar vacío el campo.

5.7.2.12. Validación por provincia

Al igual que en la validación por ciudad existen dos opciones, el usuario puede rellenar el campo con un conjunto de caracteres no numéricos o seleccionar un valor de una lista.

Esta elección del tipo de validación depende de los requisitos que el cliente da para el diseño del formulario.

5.7.2.13. Validación por población

En este caso la validación es de un conjunto de caracteres no numéricos, sigue una codificación específica dependiendo del país

5.7.2.14. Validación por código postal

Depende del país de lanzamiento de la campaña.

Por ejemplo: En España [51] el sistema de código postal consta de 5 cifras y en Portugal de 7 cifras [52].

País	Código Postal	Provincia	Zona Postal
España	5 cifras	2 primeras cifras	3 restantes
Portugal	7 cifras	4 primeras cifras	3 restantes

Tabla 5: Patrón del código postal dependiendo del país.

Como se observa en la tabla en España las 2 primeras cifras hacen referencia a la provincia mientras que en Portugal se necesitan las 4 primeras cifras y las restantes en ambos indican la zona postal.

5.7.2.15. Validación por documento de identidad

Esta validación es muy importante, aunque en la mayoría de los formularios web no suele existir este campo, ya que se ha comprobado que a los usuarios les desincentiva el hecho de tener que introducir su documento de identidad.

A pesar de ello hay clientes que necesitan este tipo de información en los registros. Por esto esta validación depende de cada país y conlleva una serie de características únicas en cada país.

5.7.2.16. Validación campo de selección

Hay validaciones que solo se necesita validar que el usuario seleccione una opción de una lista desplegable. Por ejemplo: Seleccionar el nivel de estudios.

5.7.2.17. Validación campo vacío

Hay campos que son opcionales, esto es que el usuario puede dejar el campo en blanco o no.

5.7.2.18. Validación por aviso de privacidad

Esta validación se aplica al campo de tipo *CHECKBOX*²³ en html5, que el usuario debe marcar obligatoriamente.

²³**Checkbox**, atributo de HTML5. Es una casilla de verificación.

5.7.2.19. Validación numérica

Esta validación solo se aplica para campos que se necesita que sean numéricos. Por ejemplo: Un campo que la información que se busca es saber cuánto dinero debe el cliente en tarjetas de créditos.

5.7.2.20. Validación caracteres no numéricos

Con este tipo de validación solo se aceptan letras. Este tipo de validación se da en campos como población que no requiera un estándar de codificación ISO 3166 alfa-2

5.7.2.21. Validación caracteres

Esta validación acepta todo tipo de caracteres como letras, números, símbolos, etc. Esta validación se puede encontrar en campos de un área de texto, donde el usuario escribe un mensaje de texto.

5.7.3. Enganchar las validaciones.

Mediante el módulo analizador se lee el formulario original para encontrar todos los campos que se van a validar.

Por cada campo se selecciona:

- El tipo de validación que tendrá el dato.

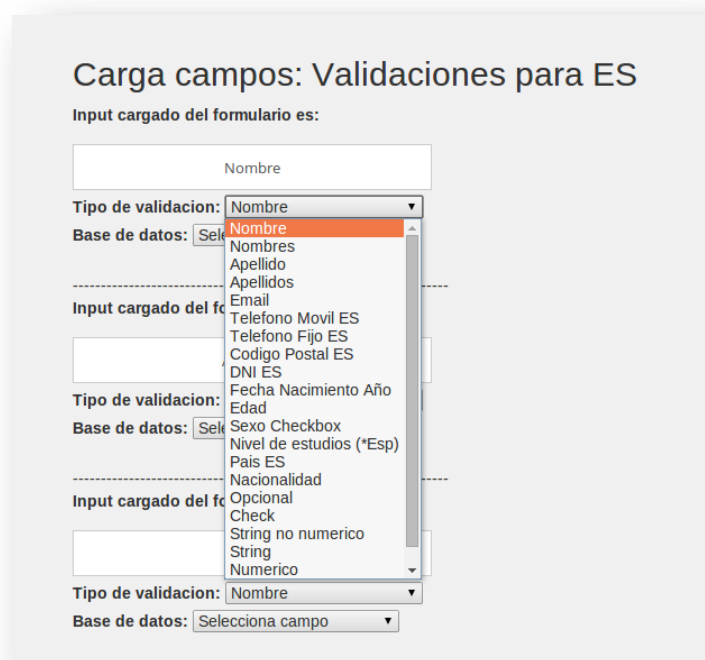


Figura 50: Ejemplo de selección de tipo de validación.

- El nombre de la columna de la base de datos donde se va a guardar la información.

Carga campos: Validaciones para ES

Input cargado del formulario es:

Nombre

Tipo de validacion: Nombre

Base de datos: TEXTO2

Input cargado: NOMBRE

Tipo de validacion: PROVINCIA

Base de datos: PROVINCIA2

Input cargado: DIRECCION

Tipo de validacion: MOVIL

Base de datos: Selecciona campo

Figura 51: Ejemplo de selección de campo de base de datos.

El programa realiza las siguientes verificaciones:

- Obliga a seleccionar el tipo de validación del campo
- Obliga a seleccionar donde guardar el dato en la base de datos.
- Comprueba que los datos no se almacene en un mismo campo de la base de datos.

5.8. Módulo modificador.

El objetivo principal de este script es sustituir los campos ya programados en el formulario original e insertar la llamada al código necesario para su funcionamiento.

Este script está programado en el lenguaje de programación Python.

5.8.1. Funcionamiento del script:

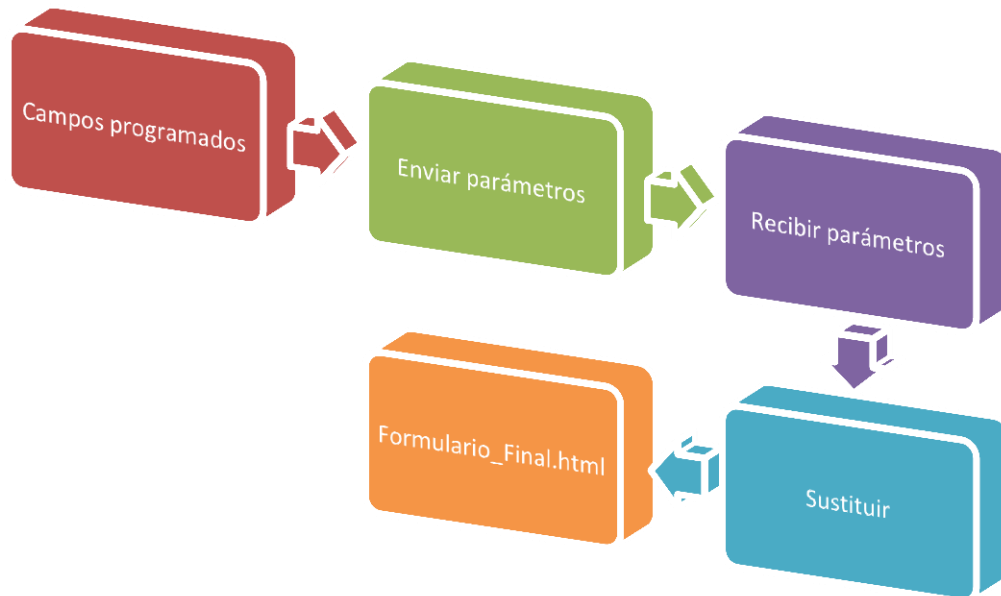


Figura 52: Flujo de trabajo del script Modificador.

5.8.1.1. Campos programados

El usuario selecciona en la aplicación dónde quiere guardar la información que introduce en el campo de una lista desplegable que contiene todas las columnas de la base de datos. También debe seleccionar en la aplicación que tipo de validación se aplica al campo.

Un ejemplo de validación de tres campos que son: Nombre, Apellidos y Teléfono.

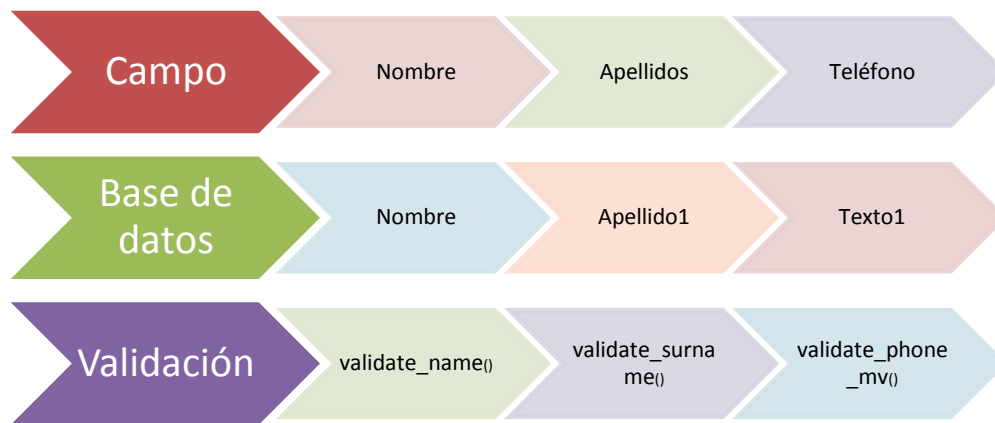


Figura 53: Ejemplo de validación de los campos de un formulario.

5.8.1.2. Envío de parámetros

Después de realizar la programación de todos los campos del formulario la aplicación envía cuatro parámetros al **script Modificar**, estos parámetros son:

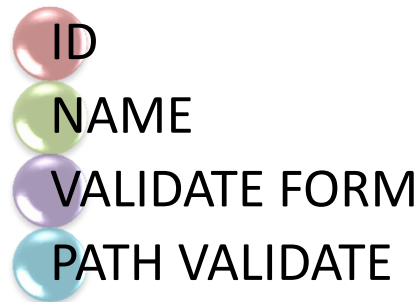


Figura 54: Parámetros que recibe el script Modificar.

El parámetro **id** contiene el nombre de la columna de la base de datos, en la que se va a guardar el dato que inserte el usuario al registrarse en el formulario.

El parámetro **name** engloba el nombre de la función de validación correspondiente al campo.

El parámetro **validate form** es el nombre de la función de validación del formulario que está en el fichero JavaScript

El parámetro **path validate** tiene la ruta donde se encuentra este fichero JavaScript que es necesario para que las validaciones de los campos ya programados funcionen correctamente.

5.8.1.3. Recibo de parámetros

El script recibe los cuatro parámetros en cuatro listas independientes separadas por un espacio. Cada lista tiene la información de cada campo, continuando con el ejemplo anterior de los campos programados, para estos campos las listas que se envían son:

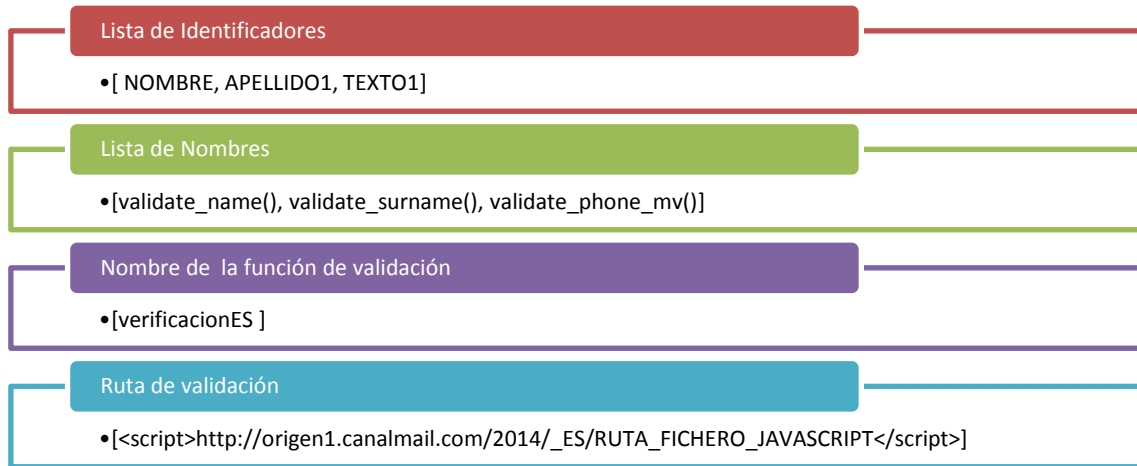


Figura 55: Ejemplo de la información que contienen los parámetros

5.8.1.4. Sustitución

Una vez que el script Modificar ha recibido la información, vuelve a leer el formulario original y reemplaza en cada campo el valor correspondiente a cada parámetro de cada lista. La sustitución se realiza tal y como se observa en el siguiente gráfico:

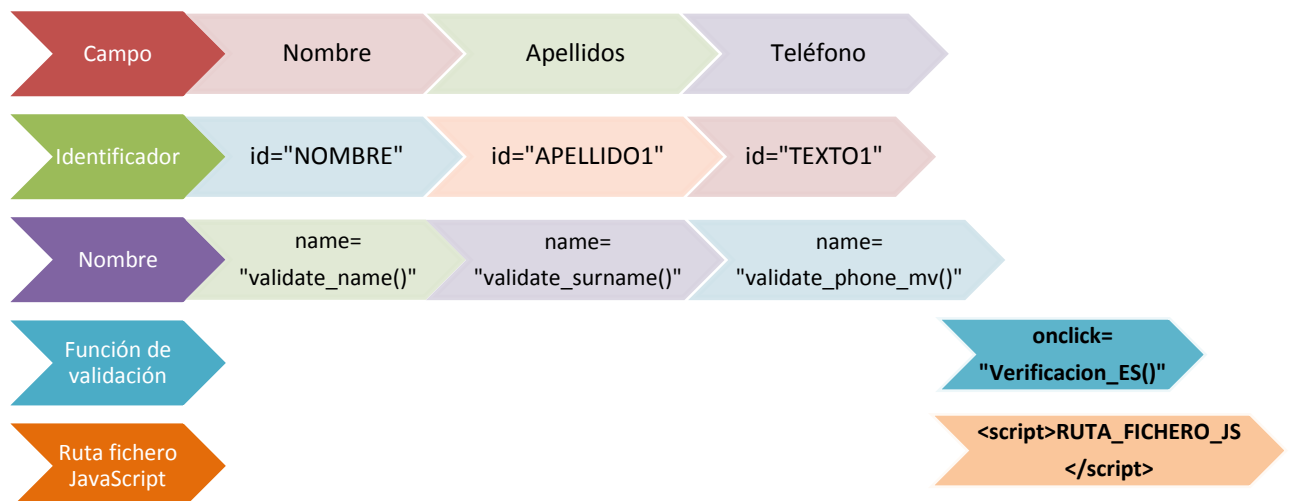


Figura 56: Ejemplo de los campos del formulario ya modificado.

5.8.1.5. *Formulario Final*

Por último después de reemplazar los parámetros en cada campo del formulario, ya está validado y listo para poder enviarse o descargarse.

Una vez que el formulario ya ha sido programado, el funcionamiento de la validación es la siguiente:

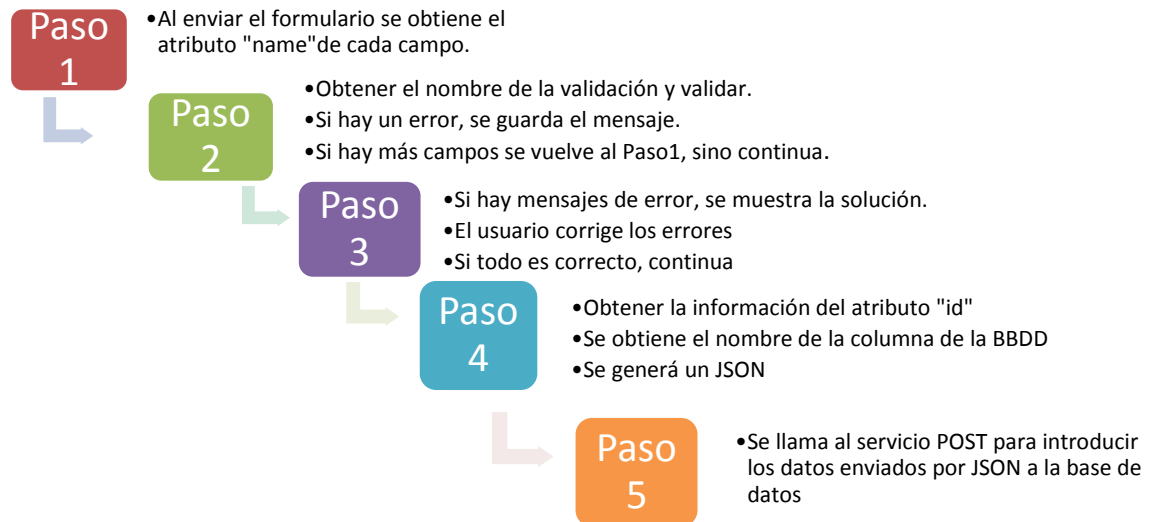


Figura 57: Funcionamiento de la validación de los campos.

5.9. Comprobación del formulario.

5.9.1. Prueba del funcionamiento.

Después de realizar el proceso de programación de los campos del formulario web, el usuario puede comprobar que las validaciones del formulario sean correctas.

Al hacer clic en el botón de probar, se visualiza el formulario ya programado tal y como se verá finalmente en un navegador web para realizar una segunda comprobación de los posibles errores de visualización y comprobar que la programación de los campos es la correcta.

Para comprobar que las validaciones que se han elegido cumplen su función, el usuario envía una prueba real rellenando los campos del formulario.

Aquí el usuario puede realizar todo tipo de pruebas para probar que las validaciones cumplen su función.

La información se almacena en la base de datos en la tabla *Ravioli* en el sistema de gestión de base de datos de MySQL, el cual comparte similitud con la base de datos de preproducción y producción en el sistema de gestión de base de datos de Oracle.

Después de introducir una prueba real, al usuario le llegará un mensaje a su correo electrónico con la información que introdujo en la prueba y podrá comprobar de esta forma que es correcto o si por el contrario hubo algún problema.

Por último, este formulario ya programado se puede descargar con las validaciones incluidas.

5.10. Sistema de Gestión de bases de datos.

5.10.1. Envío de los datos a la BBDD.

Se creó una base de datos local para guardar los datos que llegan de los formularios, luego se programó un cron que copia dichos datos a la base de datos central.

Cuando un usuario rellena un formulario, se genera un JSON²⁷ con los datos introducidos y su columna respectiva en la que se guardara.

Nombre de la columna de la BBDD	Dato introducido por el usuario
TEXT02	622123432
EMAIL1	prueba@gmail.com
NOMBRE	Aydee

Tabla 6: Ejemplo de los datos que se envían.

El ejemplo anterior con formato JSON es:

```
{“TEXT02”:”622123432” , “EMAIL1”:”prueba@gmail.com” , “NOMBRE”:”Aydee”}
```

Figura 58: El ejemplo en formato JSON.

Dicha secuencia de datos se envía mediante una llamada REST²⁸ con un método POST²⁹.

²⁷ **JSON**, término que significa “JavaScript Object Notation”, es un tipo de formato para el intercambio de datos.

²⁸ **REST**, en inglés “Representational State Transfer”. Es una técnica de arquitectura software para WWW.

²⁹ **POST**, es un método de peticiones HTTP.

5.11. Gestión de formulario.

En el menú principal de la aplicación se observa los siguientes apartados:

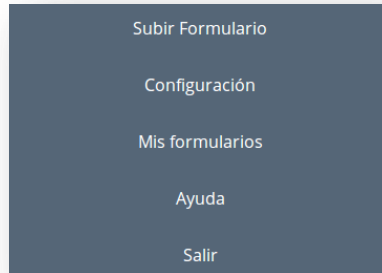


Figura 59: Opciones del menú principal.

El usuario final además de seleccionar *Subir Formulario*, puede realizar las siguientes acciones:

- *Configuración*: El usuario podrá cambiar la contraseña de acceso a la aplicación.
- *Mis Formularios*: En este apartado el usuario podrá ver y gestionar los formularios que ya han sido validados.



Figura 60: Lista de los formularios guardados en el directorio del usuario.

Cuando se visualice el formulario, el usuario tendrá la opción de descargar el fichero nuevamente.

- *Ayuda*: Aquí se encuentra la información útil del funcionamiento de la aplicación. Es decir, el manual de usuario que podrá descargarse.
- *Salir*: Al hacer clic en este apartado, se destruye la sesión del usuario y se redirige a la página de inicio.

6. VISUALIZACION

6.1. Vistas.

6.1.1. Iniciar sesión y registro.

Se presenta la vista principal de acceso a la aplicación web.

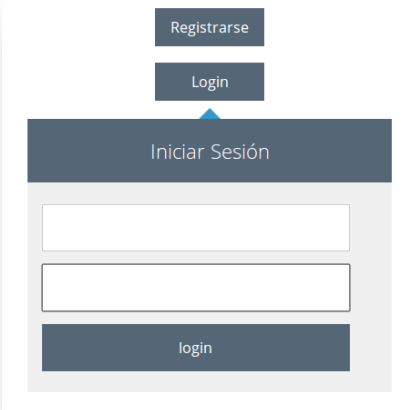


Diagrama de la interfaz de inicio de sesión. En la parte superior, hay dos botones: 'Registrarse' y 'Login'. Debajo de ellos, un botón grande y ancho con el texto 'Iniciar Sesión'. A continuación, se encuentran dos campos de entrada de texto (uno para el usuario y otro para la contraseña). Al final, un botón 'login'.

Figura 61: Vista de la página principal, siendo iniciar sesión.

En la siguiente imagen, se observa los campos obligatorios que debe rellenar el usuario para registrarse.

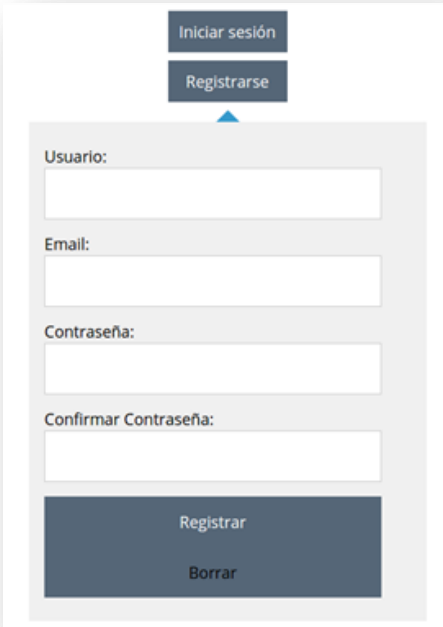


Diagrama del formulario de registro. En la parte superior, hay dos botones: 'Iniciar sesión' y 'Registrarse'. Debajo de ellos, un botón grande y ancho con el texto 'Registrarse'. A continuación, se encuentran cuatro campos de entrada de texto con las etiquetas 'Usuario:', 'Email:', 'Contraseña:', y 'Confirmar Contraseña:'. Al final, hay dos botones: 'Registrar' y 'Borrar'.

Figura 62: Vista del formulario para el registro.

6.1.2. Menú principal.

Esta imagen presenta la visualización de las opciones del menú principal.

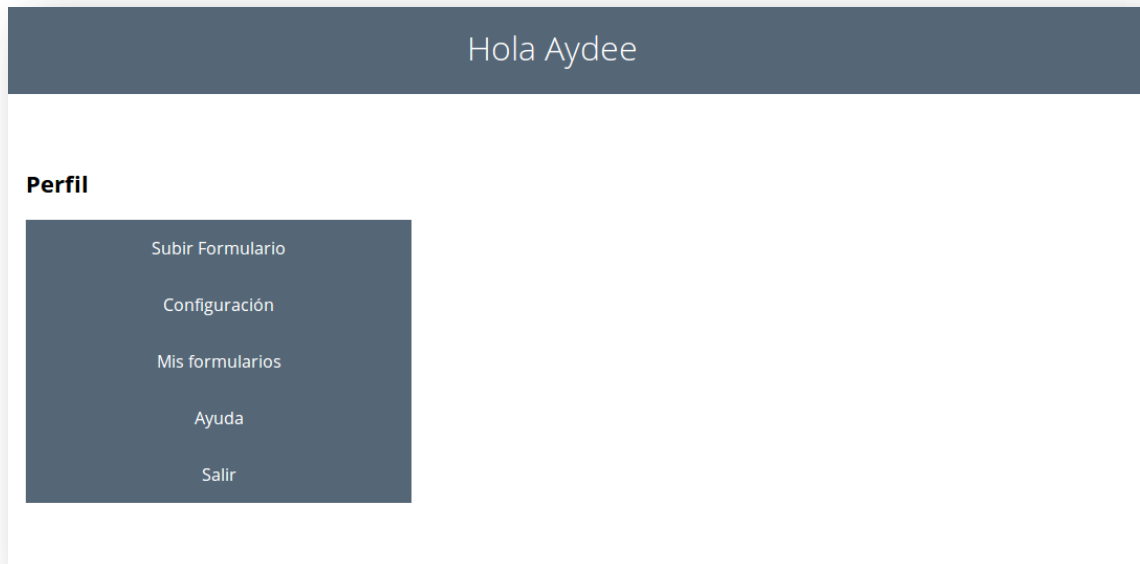


Figura 63: Vista del menú principal.

6.1.3. Subir Formulario

En la siguiente imagen se muestra la vista de subir el formulario.

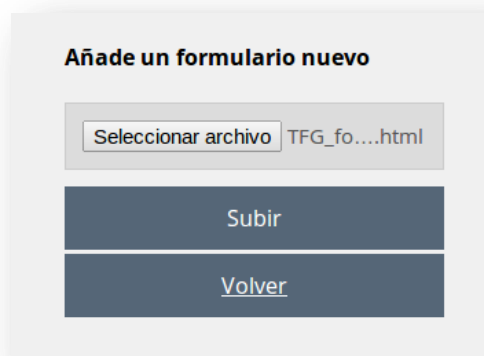
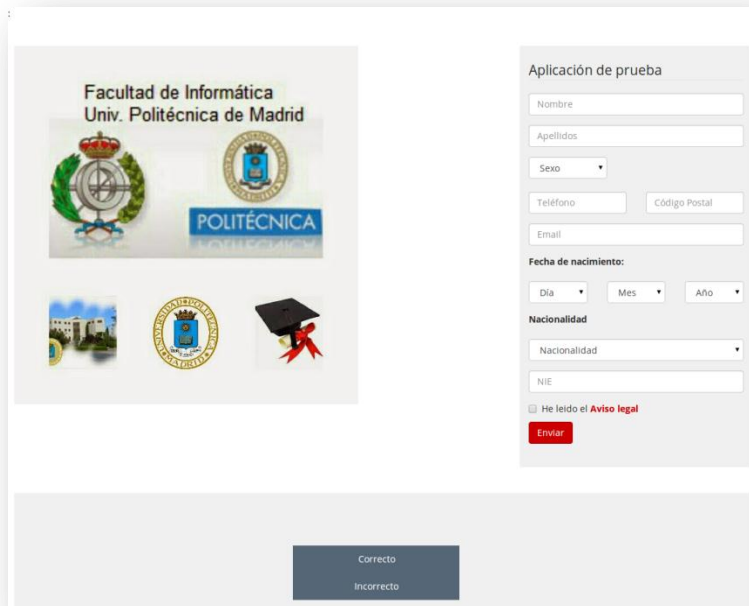


Figura 64: Vista de subir formulario.

6.1.4. Visualización del formulario.

Se observa un formulario ficticio cargado correctamente, antes de programarlo.

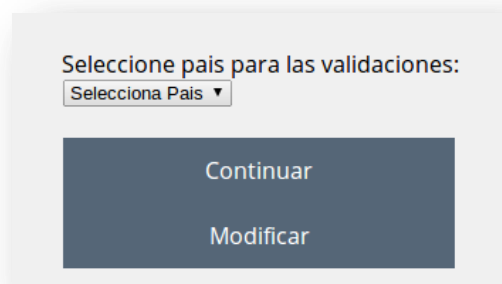


The screenshot shows a web form titled "Aplicación de prueba". On the left, there is a header area with the text "Facultad de Informática Univ. Politécnica de Madrid" and several logos, including the university's coat of arms and a "POLITÉCNICA" logo. The form itself contains the following fields: "Nombre" (text input), "Apellidos" (text input), "Sexo" (dropdown menu), "Teléfono" (text input), "Código Postal" (text input), "Email" (text input), "Fecha de nacimiento:" (with separate dropdowns for "Día", "Mes", and "Año"), "Nacionalidad" (dropdown menu), and "NIE" (text input). Below these fields is a checkbox labeled "He leído el Aviso legal" and a red "Enviar" button. At the bottom of the form, there are two buttons: "Correcto" and "Incorrecto".

Figura 65: Vista de un formulario ficticio cargado correctamente.

6.1.5. Selección de las validaciones.

Vista donde se selecciona el país de lanzamiento de la campaña.



The screenshot shows a small dialog box titled "Seleccione país para las validaciones:". It contains a dropdown menu labeled "Selecciona Pais" with a downward arrow. Below the dropdown are two buttons: "Continuar" and "Modificar".

Figura 66: Vista de la selección del país de la campaña.

Vista general de los campos de un formulario que se van a programar.

Carga campos: Validaciones para ES

Input cargado del formulario es:

Nombre

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Apellidos

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Sexo

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Telefono

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

CodPostal

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Email

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Dia

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Mes

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Año

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Nacionalidad

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Documento de identidad

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

AvisoPrivacidad

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Continuar

Figura 67: Vista de un formulario preparado para programarse.

Vista de la selección del tipo de validación.

Carga campos: Validaciones para ES

Input cargado del formulario es:

Nombre

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Apellidos

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Sexo

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Telefono

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

CodPostal

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Nacionalidad

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

Documento de identidad

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Input cargado del formulario es:

AvisoPrivacidad

Tipo de validacion: Nombre

Base de datos: Selecciona campo

Continuar

Figura 68: Lista de los tipos de validación del formulario.

Vista de la selección del campo de la base de datos.

The screenshot shows a web form titled "Carga campos: Validaciones para ES". It contains several input fields and dropdown menus. The first input field is labeled "Nombre". Below it, there is a dropdown menu for "Tipo de validacion:" with "Nombre" selected. Another dropdown menu for "Base de datos:" shows a list of fields: EMAIL, EMAIL1, EMAIL2, NOMBRE (highlighted in orange), APELLIDOS, APELLIDO1, APELLIDO2, PAIS, PROVINCIA, PROVINCIA2, POBLACION, POBLACION2, DIRECCION, DIRECCION2, SECTOR, OCUPACION, SEXO, FECHA NACIMIENTO*, MOVIL, and TELEFONO. Below this, there are more input fields and dropdown menus, including one for "Tipo de validacion:" and another for "Base de datos:" with "Selecciona campo" selected.

Figura 69: Lista de los campos de la base de datos.

6.1.6. Fin de la validación

Vista del final de la programación del formulario. Muestra las dos opciones que dispone el sistema.

The screenshot shows a white box with a message: "Su formulario ha sido programado correctamente." Below the message are two buttons: "Descargar" (blue) and "Probar" (green).

Figura 70: Vista de las opciones finales de un formulario.

6.1.7. Cambiar contraseña



The screenshot shows a web interface titled 'Configuración'. Below the title, there is a section labeled 'Cambiar la contraseña:'. This section contains a single text input field. Below the input field is a green button with the text 'Cambiar'.

Figura 71: Vista de cambiar la contraseña.

6.1.8. Lista de los formularios.

En el apartado *Mis formularios*, se muestran los formularios programados, que se encuentran ubicados en el directorio del usuario.



The screenshot shows a web interface with a dark blue header bar containing the text 'Hola Aydee'. Below the header, the title 'Lista de formularios' is displayed. The main content area contains three identical form entries, each enclosed in a box. Each entry has a title 'Formularios' and a subtitle 'Nombre del Formulario:'. The first entry's subtitle is 'form.html', the second is 'TFG_formulario.html', and the third is 'formulario2.html'. Below each subtitle are two buttons: a green 'Visualizar' button and a red 'Borrar' button.

Figura 72: Vista de la lista de formularios.

7. PRUEBAS

7.1. Pruebas.

Para comprobar el funcionamiento de la aplicación, se realizaron diferentes pruebas con usuarios reales.

7.1.1. Prueba 1: Crear una cuenta.

La primera tarea consiste crear una cuenta para poder acceder al sistema. Este apartado ha sido realizado satisfactoriamente ya que carece de dificultad.

7.1.2. Prueba 2: Descargarse el manual.

Los usuarios tendrán que descargarse el manual para entender el funcionamiento de la aplicación y de esta manera saber usarla.

Para agilizar esta prueba, además de leerse el manual, se realizó una presentación para explicar de manera sencilla como programar un formulario con la aplicación. Al final de dicha presentación se hizo una ronda de preguntas.

7.1.3. Prueba 3: Diseño de una plantilla.

Se pidió a los creativos diseñar un formulario simple siguiendo las pautas anteriormente establecidas. Esto es con todas las etiquetas necesarias para el correcto funcionamiento de la aplicación.

A un integrante del departamento creativo se le olvido insertar la etiqueta de *F_campo*, por lo que se le recordó la importancia de esta etiqueta.

7.1.4. Prueba 4: Programar el formulario.

Esta tarea es la más importante, ya que es la función principal del proyecto. Los usuarios tendrán que ser capaces de generar un formulario con las validaciones necesarias.

Al ser un formulario simple, la tarea ha sido realizada satisfactoriamente. Sin embargo, hubo varias dudas al elegir en qué campo guardar los datos.

8. CONCLUSIONES

8.1. Conclusión.

Al terminar el proyecto de desarrollo de una aplicación web empresarial de ayuda a la producción de campañas publicitarias, son diversas las conclusiones que se han obtenido, como primer punto se ha destacar el aprendizaje de nuevas herramientas web necesarias para alcanzar el objetivo final.

A lo largo del semestre he aprendido como desarrollar paso a paso esta aplicación, analizando los requisitos, sus funcionalidades mínimas y teniendo un trato cercano con el usuario final. De esta forma he conseguido realizar las tareas propuestas y cumplir con los objetivos planteados.

La aplicación se ha implementado a medida de la empresa Canalmail S.L, ofreciendo a los usuarios la posibilidad de realizar la programación de un formulario de cualquier campaña de email marketing de una forma simple, eficaz, y rápida.

Es importante mencionar los resultados obtenidos mediante las diferentes pruebas realizadas ya que se ha observado que la interfaz de usuario es intuitiva, la programación del formulario funciona correctamente y que el sistema en general cuenta con las validaciones suficientes para evitar errores por parte de los usuarios.

A pesar de que la aplicación está diseñada para unos usuarios finales concretos, los cuales son los trabajadores del departamento creativo de la empresa, creo firmemente que cualquier trabajador de otro departamento de la empresa, podría usarlo sin ningún problema, previamente explicándole el funcionamiento principal de esta aplicación.

Por esto, estoy segura que esta aplicación tiene un gran potencial de desarrollo y un futuro muy prometedor. En cuanto al diseño del mismo, se puede concluir que cumple con las condiciones necesarias para facilitar su uso al usuario.

9. FUTURAS LINEAS DE DESARROLLO

9.1. Mejoras.

Durante este semestre se realizó el proyecto siguiendo el plan de trabajo, creando la funcionalidad principal de la aplicación, sin embargo se podrían mejorar varios aspectos de los cuales destacan:

9.1.1. Volver a modificar un formulario.

Cuando se genera un nuevo formulario con sus validaciones, ya no se puede modificar el tipo de validación que se aplica a cada campo, ni la columna en la que se guarda en la base de datos.

Al listar los formularios realizados, se tendría que añadir una opción de editar para poder cambiar dichos datos.

9.1.2. Mejorar el Front-End de la aplicación.

La visualización de la aplicación es muy pobre dado que se ha contado con poco tiempo para el desarrollo de este apartado.

Se tendría que profundizar más en el diseño y cambiar las plantillas utilizadas. Al ser una aplicación con una arquitectura orientada en MVC, el cambio no supone ninguna dificultad ya que es independiente.

9.1.3. Añadir las validaciones restantes.

Se han realizado las validaciones propias de cada país donde la empresa tiene más presencia. Sin embargo en el caso de Latinoamérica se podría profundizar en la clasificación de las validaciones respectivas para cada país.

9.1.4. Crear un panel de administrador.

Otra mejora que se podría realizar es crear un panel administrador. En este panel se podría gestionar el cambio de contraseña, la visualización de los formularios programados de cada usuario y tener la posibilidad de descargarlos.

10. BIBLIOGRAFÍA

10.1. Bibliografía.

- [1] MediaResponse Team. (2015) Mediarresponse. [Online]. <http://www.mediarresponse.com/aboutus.html>
- [2] MediaResponse Team. (2015) Hotwords. [Online]. <http://www.hotwords.com/>
- [3] MediaResponse Team. (2015) [Online]. <http://www.webzodes.com/>
- [4] MediaResponse Team. (2013) Efficient Target. [Online]. <http://www.efficienttarget.com/>
- [5] MediaResponse Team. (2014) CanalMail. [Online]. <http://www.canalmail.com/>
- [6] Wikipedia, La comunidad de. (2014, Diciembre) Wikipedia. [Online]. http://es.wikipedia.org/wiki/Formulario_web
- [7] Wikipedia, La comunidad de. (2014, Diciembre) Wikipedia. [Online]. http://es.wikipedia.org/wiki/Apache_Struts
- [8] thaicreate. [Online]. <http://www.thaicreate.com/images/java-struts-tutorial.jpg>
- [9] Juan Ardisson. (2012, Febrero) maestrosdelweb. [Online]. <http://www.maestrosdelweb.com/curso-symfony2-introduccion-instalacion/>
- [10] Symfony. (2015) Symfony. [Online]. <http://symfony.es/que-es-symfony>
- [11] Symfony. (2014) Symfony. [Online]. http://symfony.com/logos/symfony_black_03.png
- [12] La comunidad de Wikipedia. (2014, Julio) Wikipedia. [Online]. http://es.wikipedia.org/wiki/Zend_Framework
- [13] Zend. (2015) Zend Framework. [Online]. <http://framework.zend.com/images/logos/zf-logo-mark.png>
- [14] Julio Pari. (2011, Mayo) Blog juliopari. [Online]. <http://blog.juliopari.com/symfony-vs-zend-framework/>
- [15] La comunidad Wikipedia. (2015, Enero) Wikipedia. [Online]. <http://es.wikipedia.org/wiki/NetBeans>
- [16] Marina Sprava. (2013, Mayo) Blog jelastic. [Online]. <http://blog.jelastic.com/wp-content/uploads/2013/04/netbeans-logo.jpg>

- [17] La comunidad Wikipedia. (2014, Noviembre) Wikipedia. [Online]. <http://es.wikipedia.org/wiki/HTML5>
- [18] La comunidad Wikipedia. (2014, Diciembre) Wikipedia. [Online]. <http://es.wikipedia.org/wiki/JavaScript>
- [19] Asociacion aepi. (2014) Asociacionaepi. [Online]. <http://www.asociacionaepi.es/wp-content/uploads/2013/12/JavaScriptLogo.png>
- [20] Guillermo Villaverde. (2013, Abril) Convirtiendote.pro. [Online]. <http://convirtiendote.pro/javascript/filtrando-arreglos-y-objetos-con-jquery/>
- [21] Mootools. (2015) Mootools. [Online]. <http://mootools.net/>
- [22] La comunidad de Wikipedia. (2014, Octubre) Wikipedia. [Online]. <http://es.wikipedia.org/wiki/JQuery>
- [23] La comunidad de Wikipedia. (2013, Julio) Wikipedia. [Online]. <http://es.wikipedia.org/wiki/Mootools>
- [24] Techwars. (2014) Techwars. [Online]. <http://www.techwars.io/fight/jquery/mootools/>
- [25] Aaron Newton. (2009, Mayo) JQueryvsmootools. [Online]. http://www.jqueryvsmootools.com/index_es-ar.html
- [26] Percy Alex. (2014, Noviembre) Itechnode. [Online]. <http://itechnode.com/bootstrap-3-el-manual-oficial>
- [27] Alvaro Fontela Sanchez. (2014) Openwebcms. [Online]. <http://openwebcms.es/2013/que-es-bootstrap/>
- [28] (2014) taylorssites. [Online]. <http://www.taylorssites.com/wp-content/uploads/2013/01/foundation-logo.jpg>
- [29] Zurb. (2015) Foundation Zurb. [Online]. <http://foundation.zurb.com/index.html>
- [30] Vermilion. (2014) Vermilion. [Online]. <http://responsive.vermilion.com/compare.php>
- [31] La comunidad de Wikipedia. (2014) Wikipedia. [Online]. <http://es.wikipedia.org/wiki/PHP>
- [32] Yair Rodríguez. (2014, Septiembre) Duilio. [Online]. <http://duilio.me/como-instalar-lamp-en-ubuntu-linux/>

- [33] Zkoss. (2014) Zkoss. [Online]. <http://www.zkoss.org/product/zkjsp>
- [34] Javier A. (2014) Chuidiang. [Online]. <http://blog.chuidiang.com/2009/04/23/%C2%BFjsp-o-php/>
- [35] La comunidad Wikipedia. (2015, Enero) Wikipedia. [Online]. <http://es.wikipedia.org/wiki/Python>
- [36] Python. (2015) Python. [Online]. <https://www.python.org/>
- [37] Viapeer. (2015) Viapeer. [Online]. <http://viapeer.com/services/info.php?service=perl>
- [38] La comunidad Wikipedia. (2014, Diciembre) Wikipedia. [Online]. <http://es.wikipedia.org/wiki/Perl>
- [39] La comunidad Wikipedia. (2014, Abril) Wikipedia. [Online]. http://es.wikipedia.org/wiki/Algoritmo_de_b%C3%BAsqueda_de_cadenas_Boyer-Moore
- [40] busquedasecuencialdetexto. Google Sites. [Online]. <https://sites.google.com/site/busquedasecuencialdetexto/algoritmo-boyer-moore>
- [41] La comunidad Wikipedia. (2014, Noviembre) Wikipedia. [Online]. http://es.wikipedia.org/wiki/Algoritmo_Knuth-Morris-Pratt
- [42] MySQL. (2015) MySQL. [Online]. <http://www.mysql.com/>
- [43] PostgreSQL. (2013) PostgreSQL. [Online]. <http://www.postgresql.org/es/>
- [44] SQLite. (2014) SQLite. [Online]. <http://www.sqlite.org/>
- [45] Progress. (2015) Progress. [Online]. <https://www.progress.com/products/openedge/features/operational-excellence/database>
- [46] IBM. (2014) IBM. [Online]. <http://www-01.ibm.com/software/data/informix/>
- [47] Mediarresponse. (2014, Enero) Mediarresponse Blog. [Online]. <http://blog.mediarresponse.com.br/es/presentamos-al-departamento-creativo-mediarresponse/>
- [48] Daniel G. (2012, Febrero) Gorditosybonitos. [Online]. <http://www.gorditosybonitos.com/los-prefijos-de-telefono-en-mexico/>

- [49] La comunidad Wikipedia. (2014, Julio) Wikipedia. [Online].
http://es.wikipedia.org/wiki/Anexo:Prefijos_telef%C3%B3nicos_de_Espa%C3%B1a
- [50] La comunidad Wikipedia. (2014, Diciembre) Wikipedia. [Online].
http://es.wikipedia.org/wiki/ISO_3166-1
- [51] La comunidad Wikipedia. (2014, Agosto) Wikipedia. [Online].
http://es.wikipedia.org/wiki/C%C3%B3digo_postal_espa%C3%B1ol
- [52] La comunidad Wikipedia. (2014, Septiembre) Wikipedia. [Online].
http://en.wikipedia.org/wiki/Postal_codes_in_Portugal

A. ANEXO

A.1. Manual de usuario.

En este manual se explica cómo realizar las diferentes funciones que nos ofrece la aplicación web.

A.1.1. Registro e inicio de sesión.

Al ingresar en la aplicación, aparece la página de inicio.



Registrarse

Login

Iniciar Sesión

aydee

.....

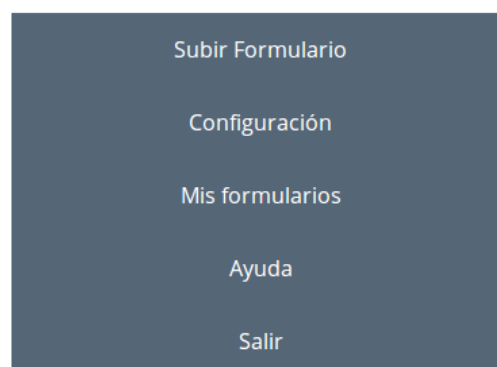
login

Figura 73: Página de inicio.

Para poder crear una nueva cuenta se hace clic en el botón “Registrarse”. Si ya se posee una cuenta, se introduce el nombre de usuario y la contraseña, y se hace clic en “Login”.

Al ingresar en la aplicación, aparece el siguiente menú:

Perfil



Subir Formulario

Configuración

Mis formularios

Ayuda

Salir

Figura 74: Menú del usuario.

A.1.2. Subir Formulario.

Para generar un formulario, se hace clic en “subir formulario”. En la siguiente pantalla se pedirá cargar el formulario que se quiere validar.

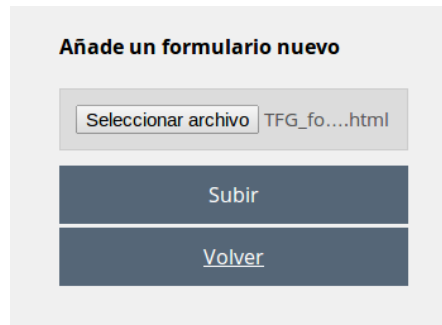


Figura 75: Servicio para cargar el nuevo formulario.

A.1.3. Visualización del formulario.

Al cargar el formulario, se mostrara este para asegurarnos que el diseño es correcto.

Hay dos botones al final de la página:

- Correcto: Se hace clic en este botón si el formulario se ve correctamente.
- Incorrecto: Si se ve algún error en la visualización del diseño del formulario, se hace clic en este botón, y aparecerá una guía de los errores más frecuentes y como solucionarlo.

Comprueba que el formulario se ve correctamente:
 - Imágenes se cargan correctamente
 - Se lee correctamente



Figura 76: Visualización del formulario.

A.1.4. Selección del País para las validaciones.

Luego, se tendrá que seleccionar el país al que está destinada la campaña publicitaria.

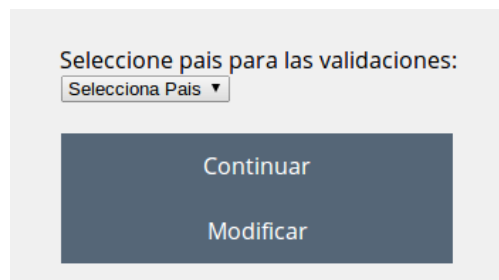


Figura 77: Selección del país para las validaciones.

A.1.5. Validaciones y campos de la BBDD.

Y finalmente para cada campo del formulario se seleccionará un tipo de validación, y el campo de la base de datos en el que se desea guardar el dato.

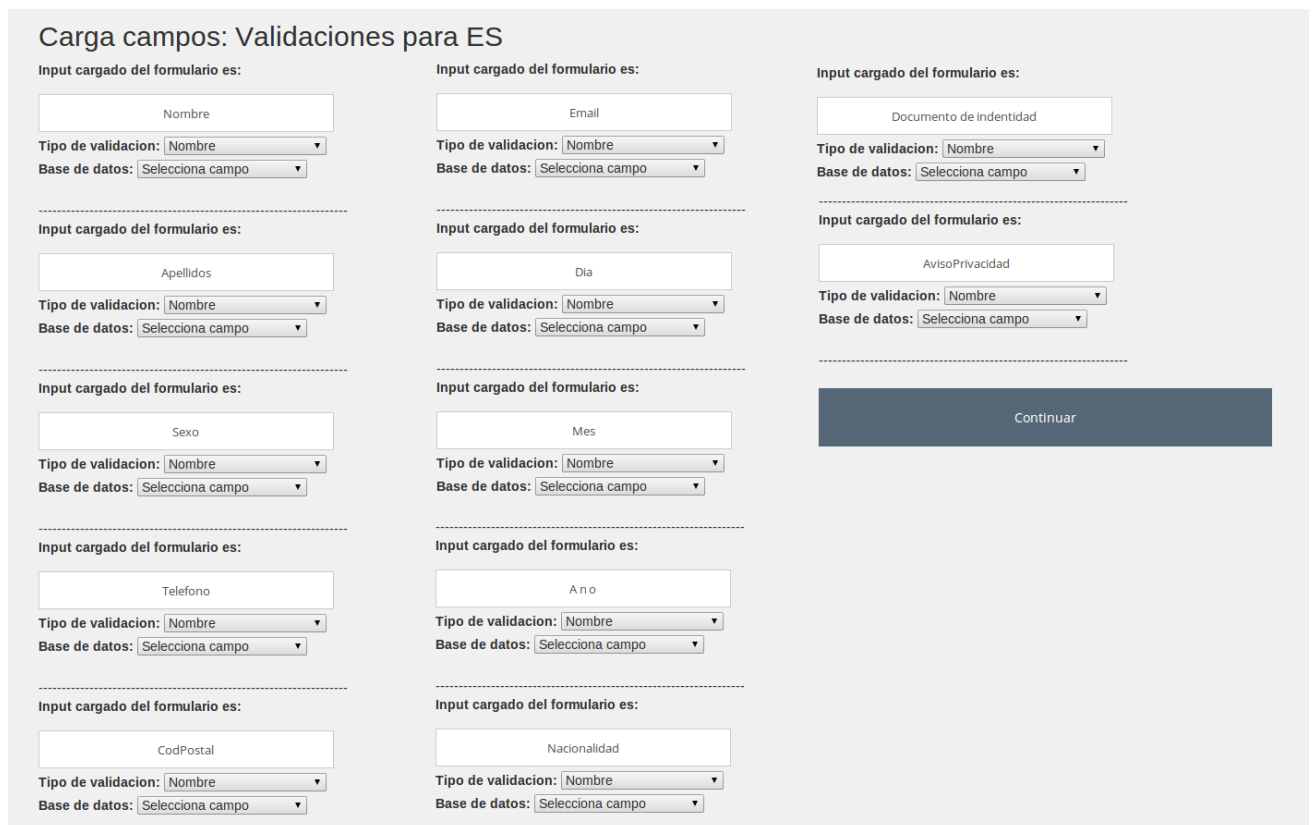


Figura 78: Seleccionar las validaciones y los campos de la BBDD.

Nota: No se puede seleccionar 2 veces el mismo campo de base de datos.

Al generar el formulario, se da la posibilidad de descargarlo o de probarlo online.

Su formulario ha sido programado correctamente.



Figura 79: Finalización de las validaciones.

A.1.6. Configuración.

Se puede modificar la contraseña en el apartado de Configuración.

Configuracion

Cambiar la contraseña:

Cambiar

Figura 80: Configuración.

A.1.7. Ver formularios.

Para poder ver todos los formularios que se han generado, se hace clic en “Mis formularios” situado en el menú principal.

Lista de formularios

Formularios
Nombre del Formulario: form.html
Visualizar Borrar

Formularios
Nombre del Formulario: TFG_formulario.html
Visualizar Borrar

Formularios
Nombre del Formulario: formulario2.html
Visualizar Borrar

Figura 81: Lista de formularios programados.

A.2. Campos de la BBDD.

A.2.1 Esquema de la BBDD.

Nombre	Tipo
EMAIL	VARCHAR2(100)
EMAIL2	VARCHAR2(100)
NOMBRE	VARCHAR2(45 CHAR)
IDENT	VARCHAR2(45 CHAR)
APELLIDOS	VARCHAR2(60 CHAR)
EMPRESA	VARCHAR2(255 CHAR)
PAIS	VARCHAR2(255 CHAR)
PROVINCIA	VARCHAR2(255 CHAR)
POBLACION	VARCHAR2(255 CHAR)
DIRECCION	VARCHAR2(255 CHAR)
SECTOR	VARCHAR2(255 CHAR)
OCUPACION	VARCHAR2(255 CHAR)
SEXO	VARCHAR2(1)
FEC_NCTO	DATE
MOVIL	VARCHAR2(15 CHAR)
TELEFONO	VARCHAR2(15 CHAR)
CP	VARCHAR2(20 CHAR)
PED_COD	NUMBER(8)
FEC_INS	DATE
IP	VARCHAR2(255)
APELLIDO2	VARCHAR2(60 CHAR)
EMAIL1	VARCHAR2(100)
CARGO	VARCHAR2(60 CHAR)
ESTADO_CIVIL	VARCHAR2(1)
TIPO_VIA	VARCHAR2(4 CHAR)
NUMERO_DIRECCION	VARCHAR2(8 CHAR)
HORA_CONTACTO	VARCHAR2(50 CHAR)
FORMA_CONTACTO	VARCHAR2(1 CHAR)
LEAD_ID	NUMBER(14)
RESTO_DIRECCION	VARCHAR2(16 CHAR)
CHECKBOX1	VARCHAR2(1 CHAR)
CHECKBOX2	VARCHAR2(1 CHAR)
CHECKBOX3	VARCHAR2(1 CHAR)
CHECKBOX4	VARCHAR2(1 CHAR)
CHECKBOX5	VARCHAR2(1 CHAR)
CHECKBOX6	VARCHAR2(1 CHAR)
TEXT01	CLOB
CUENTA	VARCHAR2(20 CHAR)
BANCO	VARCHAR2(30 CHAR)
VALIDO	VARCHAR2(1)
ASO_COD	VARCHAR2(60 CHAR)
RED	NUMBER(6)

HASH_REGISTRO	NUMBER(10)
URL_REFERER	VARCHAR2(400 CHAR)
TITULAR	VARCHAR2(60 CHAR)
TARJ_NUMERO	VARCHAR2(16 CHAR)
TARJ_CADUCIDAD	VARCHAR2(4 CHAR)
TAR_SEGURIDAD	VARCHAR2(6 CHAR)
OPCION	VARCHAR2(12 CHAR)
FAX	VARCHAR2(24 CHAR)
NIVEL_VALIDACION	VARCHAR2(1 CHAR)
PISO_DIRECCION	VARCHAR2(4 CHAR)
PUERTA_DIRECCION	VARCHAR2(4 CHAR)
ESCALERA_DIRECCION	VARCHAR2(4 CHAR)
BLOQUE_DIRECCION	VARCHAR2(4 CHAR)
CANAL	VARCHAR2(120 CHAR)
CANAL_REFERRER	VARCHAR2(120 CHAR)
DIRECCION_ENVIO	VARCHAR2(800 CHAR)
FORMA_PAGO	VARCHAR2(200 CHAR)
TARJ_TIPO	VARCHAR2(40 CHAR)
DIRECCION2	VARCHAR2(255 CHAR)
POBLACION2	VARCHAR2(255 CHAR)
PROVINCIA2	VARCHAR2(255 CHAR)
CP2	VARCHAR2(20 CHAR)
TELEFONO2	VARCHAR2(15 CHAR)
TEXT02	VARCHAR2(200 CHAR)
TEXT03	VARCHAR2(200 CHAR)
TEXT04	VARCHAR2(200 CHAR)
TEXT05	VARCHAR2(200 CHAR)
TEXT06	VARCHAR2(200 CHAR)
TARJ_TITULAR	VARCHAR2(60 CHAR)
APELLIDO1	VARCHAR2(60 CHAR)
ID_COOKIE	NUMBER(14)
ORIGEN	VARCHAR2(10 CHAR)
PLATAFORMA	VARCHAR2(10 CHAR)
ASO_COD_ORIGINAL	VARCHAR2(60 CHAR)
RED_ORIGINAL	NUMBER(6)
ESTADO_INTEGRACION	VARCHAR2(6)
FECHA_INTEGRACION	DATE
RESPUESTA_INTEGRACION	VARCHAR2(1024)

A.3. Tipo de validaciones.

VALIDACIONES CAMPO	OBLIGATORIO	TIPO	VALIDACION	JS
PAIS	SI	SELECT	ES = España MX = México PT = Portugal LT = Latinoamérica	validate_country()
NOMBRE	*	INPUT	NOMBRE = 1 NOMBRE NOMBRES = 2 NOMBRES	validate_name()
APELLIDO	*	INPUT	APELLIDO = 1 APELLIDO APELLIDOS = 2 APELLIDOS	validate_surname()
EMAIL	*	INPUT	EMAIL = EMAIL	validate_email()
TELEFONO	*	INPUT	ES = MOVIL/FIJO MX = LADA/MOVIL/FIJO PT = MOVIL/FIJO LT = TFN NUMERICO c_code = country code +34/0034	validate_phone_fj() validate_phone_mv() validate_mx_lada() validate_mx_phone_fj() validate_mx_phone_mv() validate_pt_phone_fj() validate_pt_phone_mv() validate_lt_phone() validate_country_code()
F.NACIMIENTO	*	INPUT SELECT	EDAD DIA MES	validate_age() validate_dd() validate_mm()

			AÑO	validate_yyyy()
SEXO	*	INPUT	M = HOMBRE/F = MUJER	validate_gender1()
		CHECKBOX	M = HOMBRE/F = MUJER	validate_gender3()
NIVEL ESTUDIOS	*	SELECT	1 = BACH 2= FTP ...	validate_education()
SIT. LABORAL	*	SELECT	1 =..	validate_work_situation()
CP	*	INPUT	ES = 5 DIGITOS	validate_cp()
			MX = 5 DIGITOS	validate_mx_cp()
			PT	validate_pt_cp()
			LT	validate_lt_cp()
CALLE	*	INPUT		validate_address_street1()
		SELECT	AVENIDA/CALLE/PASAJE...	validate_address_street2()
NUMERO	*	INPUT		validate_address_number1()
		SELECT	1...9	validate_address_number2()
PISO	*	INPUT	1, 2...	validate_address_floor()
PUERTA	*	INPUT	1, A,	validate_address_door()
EXTRA	*	INPUT	Escalera B	validate_address_extra()
CIUDAD	*	INPUT	ES	validate_address_city1()
		SELECT	ES	validate_address_city2()
		INPUT	MX	validate_address_mx_city1()
		SELECT	MX	validate_address_mx_city2()
		INPUT	PT	validate_address_pt_city1()
		SELECT	PT	validate_address_pt_city2()
		INPUT	LT	validate_address_lt_city1()
		SELECT	LT	validate_address_lt_city2()

PROVINCIA	*	INPUT		validate_address_province1()
		SELECT		validate_address_province2()
		INPUT	MX	validate_address_mx_province1()
		SELECT	MX	validate_address_mx_province2()
		INPUT	PT	validate_address_pt_province1()
		SELECT	PT	validate_address_pt_province2()
		INPUT	LT	validate_address_lt_province1()
		SELECT	LT	validate_address_lt_province2()
POBLACION	*	INPUT		validate_address_poblacion1()
		SELECT		validate_address_poblacion2()
		INPUT	MX	validate_address_mx_poblacion1()
		SELECT	MX	validate_address_mx_poblacion2()
		INPUT	PT	validate_address_pt_poblacion1()
		SELECT	PT	validate_address_pt_poblacion2()
		INPUT	LT	validate_address_lt_poblacion1()
		SELECT	LT	validate_address_lt_poblacion2()
DIRECCION	*	INPUT	C/ Caraba nº11	validate_address()
PAIS	*	SELECT	COUNTRY Iso code:ES/AR...	validate_countryisocode()
ID CARD	*	INPUT	ES = DNI/NIE	validate_idcard_type()
			MX =	validate_mx_idcard_type()
			PT	validate_pt_idcard_type()
			LT	validate_lt_idcard_type()
NACIONALIDAD	*	SELECT	COUNTRY Iso code:ES/AR...	validate_nacionality()
CONTACTO HORARIO	*	SELECT	1= 10:00 -15:00 etc...	validate_contactshedule()
AVISO PRIVACIDAD	*	CHECKBOX	1: si 0:no	validate_optin_legal()
CHECK	*	CHECKBOX	1: si 0:no	validate_optin_new()

VACIO
NUMERICO

validate_empty()
validate_numeric()

Nota: Todos los campos marcados con * pueden ser obligatorios o no dependiendo de los requisitos de la campaña publicitaria.

Este documento esta firmado por



Firmante	CN=tfgm.fi.upm.es, OU=CCFI, O=Facultad de Informatica - UPM, C=ES
Fecha/Hora	Wed Jan 07 22:43:04 CET 2015
Emisor del Certificado	EMAILADDRESS=camanager@fi.upm.es, CN=CA Facultad de Informatica, O=Facultad de Informatica - UPM, C=ES
Numero de Serie	630
Metodo	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)